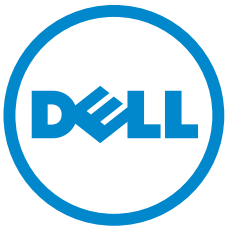
53-1002113-01
02 November 2010



PowerConnect B-MLXe

Configuration Guide

Information in this document is subject to change without notice.
© 2010 Dell Inc. All rights reserved.

Reproduction of these materials in any manner whatsoever without the written permission of Dell Inc. is strictly forbidden.

Trademarks used in this text: *Dell*, the *DELL* logo, *Inspiron*, *Dell Precision*, *Dimension*, *OptiPlex*, *Latitude, PowerEdge, PowerVault, PowerApp, Dell OpenManage* and the *YOURS IS HERE* logo are trademarks of Dell Inc.; *Intel*, *Pentium,* and *Celeron* are registered trademarks of Intel Corporation in the U.S. and other countries; *Microsoft*, *Windows*, *Windows Server*, *MS-DOS* and *Windows Vista* are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

Regulatory Model Codes: Brocade MLXe-4, Brocade MLXe-8, Brocade MLXe-16

Contents

About This Document

Audience lvii
Supported features lvii
Document conventions lxiii
Text formatting lxiii
Command syntax conventions lxiv
Notes, cautions, and danger notices lxiv
Related publications lxiv
Getting technical help or reporting errors lxv
Contacting Dell lxv

Chapter 1 Getting Started with the Command Line Interface

Overview 1
Logging on through the CLI 2
On-line help 2
Command completion 2
Scroll control 2
Line editing commands 3
EXEC commands 4
Global level 4
CONFIG commands 5
Accessing the CLI 7
Single user in CONFIG mode 8
Multi-user conflict during deletion of group configuration (or stanza) 9
Navigating among command levels 9
CLI command structure 10
Searching and filtering output 11
Allowable characters for LAG names 15
CLI parsing enhancement 16
Syntax shortcuts 16
Saving configuration changes 16

Chapter 2 Securing Access to Management Functions

Overview 17
Securing access methods 18
Restricting remote access to management functions 20
Using ACLs to restrict remote access 20

Defining the console idle time 22
Restricting remote access to the device to specific IP addresses 23
Defining the Telnet idle time 24
Specifying the maximum login attempts for Telnet access 24
Restricting remote access to the device to specific VLAN IDs 25
Enabling specific access methods 26
Setting passwords 26
Setting a Telnet password 27
Setting passwords for management privilege levels 27
Recovering from a lost password 29
Displaying the SNMP community string 30
Disabling password encryption 30
Specifying a minimum password length 30
Setting up local user accounts 31
Configuring a local user account 31
Enhancements to username and password 33
Enabling enhanced user password combination requirements 33
Requirement to accept the message of the day 36
Enhanced login lockout 36
Setting passwords to expire 36
Creating an encrypted all-numeric password 37
Granting access by time of day 37
Configuring SSL security for the Web Management Interface 37
Enabling the SSL server on a PowerConnect router 37
Importing digital certificates and RSA private key files 38
Generating an SSL certificate 38
Configuring TACACS or TACACS+ security 39
How TACACS+ differs from TACACS 39
TACACS or TACACS+ authentication, authorization, and accounting 39
TACACS or TACACS+ configuration considerations 43
Enabling SNMP traps for TACACS 44
Identifying the TACACS or TACACS+ servers 44
Specifying different servers for individual AAA TACACS functions 45
Setting optional TACACS or TACACS+ parameters 46
Configuring authentication-method lists for TACACS or TACACS+ 47
Configuring TACACS+ authorization 49
Configuring TACACS+ accounting 52
Configuring an interface as the source for all TACACS or TACACS+ packets 53
Displaying TACACS or TACACS+ statistics and configuration information 54
Configuring RADIUS security 55

RADIUS authentication, authorization, and accounting 55
RADIUS configuration considerations. 60
RADIUS configuration procedure 60
Configuring Dell-specific attributes on the RADIUS server 60
Enabling SNMP traps for RADIUS 62
Identifying the RADIUS server to the PowerConnect router . . . 63
Specifying different servers for individual AAA functions 63
Setting RADIUS parameters 64
Configuring authentication-method lists for RADIUS. 65
Configuring RADIUS authorization 67
Configuring RADIUS accounting 68
Configuring an interface as the source for all RADIUS packets 70
Displaying RADIUS configuration information 70
Configuring AAA on the console 72
Configuring AAA authentication-method lists for login 72
Configuring authentication-method lists 73
Configuration considerations for authentication-method lists 74
Examples of authentication-method lists. 75

Chapter 3 Configuring Basic Parameters

Overview 77
Entering system administration information 78
Configuring Simple Network Management (SNMP) traps 78
Specifying an SNMP trap receiver 79
Specifying a single trap source 80
Setting the SNMP trap holddown time 80
Disabling SNMP traps 81
Disabling Syslog messages and traps for CLI access. 82
Configuring SNMP ifIndex. 83
On NetIron MLX Series devices only 83
Configuring optical monitoring 84
Displaying media information. 85
Optics compatibility checking. 86
Disabling transceiver type checking 87
Designating an interface as the packet source 87
Configuring an interface as the source for all Telnet packets 87
Cancelling an outbound Telnet session 88

Configuring an interface as the source for all SSH packets 88
Configuring an interface as the source for all SNTP packets 88
Configuring an interface as the source for all TFTP packets 89
Configuring an interface as the source for all TACACS or TACACS+ packets 89
Configuring an interface as the source for all RADIUS packets 90
Specifying a Simple Network Time Protocol (SNTP) server 90
Configuring the device as an SNTP server 92
Displaying SNTP server information 93
Setting the system clock 94
DST “change" notice for networks using US time zones 95
Creating a command alias 96
Removing an alias 96
Displaying a list of all configured alias 96
Limiting broadcast, multicast, or unknown unicast rates 96
Limiting broadcasts 97
Limiting multicasts 97
Limiting unknown unicasts 97
Configuring CLI banners 97
Setting a message of the day banner 98
Setting a privileged EXEC CLI level banner 98
Displaying a message on the console when an incoming Telnet session Is detected 98
Configuring terminal display 99
Checking the length of terminal displays 99
Enabling or disabling routing protocols 99
Displaying and modifying default settings for system parameters 100
Enabling or disabling layer 2 switching 104
Configuring static MAC addresses 105
Changing the MAC age time 106
Configuring static ARP entries 106
Configuring system max values 106
Configuring CAM size for an IPv4 multicast group 109
Configuring CAM size for an IPv6 multicast group 110
Configuring profiles with a zero-size IPv4 or IPv6 ACL 111

Maintaining system-max configuration with available system resources 112
Configuration time 112
Bootup time 112
L2 elements 114
L3 elements 114
VPLS elements 115
Miscellaneous elements 115
Monitoring dynamic memory allocation 115
Switch fabric fault monitoring 117
Displaying switch fabric information 117
Displaying switch fabric module information 118
Powering a switch fabric link on or off manually 118
Powering a switch fabric module off automatically on failure 119
Switch fabric log messages 119
Switch fabric utilization monitoring 120
Verifying an image checksum 121
Displaying information for an interface for an Ethernet port 121
Displaying the full port name for an Ethernet interface 122
Displaying statistics information for an Ethernet port 125
Monitoring Ethernet port statistics in real time 125
Displaying recent traffic statistics for an Ethernet port 129
Configuring SNMP to revert ifType to legacy values 131
Configuring snAgentConfigModuleType to return original values 131
Preserving interface statistics in SNMP 132

Chapter 4 Configuring Interface Parameters

Overview 133
Assigning a port name 134
Assigning an IP address to a port 134
Modifying port speed 134
Modifying port mode 135
Auto Negotiation Speed Limit 135
Disabling or re-enabling a port 136
Disabling Source Address Learning on a port 137
Changing the default Gigabit negotiation mode 137
Changing the negotiation mode 137
Disabling or re-enabling flow control 137
Specifying threshold values for flow control 138

Modifying port priority (QoS) 139
Setting IP VPN packets with a TTL value of 1 to be dropped 139
Port transition hold timer 139
Port flap dampening 140
Configuring port link dampening on an interface 140
Configuring port link dampening on a LAG 140
Re-enabling a port disabled by port link dampening. 140
Displaying ports configured with port link dampening 141
Port loop detection 142
Strict mode and Loose mode 142
Recovering disabled ports. 142
Disable duration and loop detection interval. 142
Enabling loop detection 143
Configuring a global loop detection interval 144
Configuring the device to automatically re-enable ports. 144
Clearing loop-detection 144
Displaying loop-detection information 145
Syslog message 145
Mirroring and Monitoring 146
Configuration guidelines for monitoring traffic 146
Assigning a mirror port and monitor ports 146
Displaying mirror and monitor port configuration 147
ACL-based inbound mirroring 148
Considerations when configuring ACL-based inbound mirroring 148
Configuring ACL-based inbound mirroring 149
10G WAN PHY fault and performance management. 152
Wait for all cards feature 156
Link fault signaling 157
Displaying and clearing remote fault counters 158
Limits and restrictions 159
Local fault event detection and counters 159
Displaying and clearing local fault counters 159
Displaying Network Processor statistics 160
Relationships between some counters 163
Clearing the NP statistics counters 163

Chapter 5 Enabling the Foundry Discovery Protocol (FDP) and Reading Cisco Discovery Protocol (CDP) Packets

Overview 165
Using FDP 165
Configuring FDP 166
Displaying FDP information 167
Clearing FDP and CDP information 169

Reading CDP packets 170
Enabling interception of CDP packets globally 170
Enabling interception of CDP packets on an interface 170
Displaying CDP information 170
Clearing CDP information 172

Chapter 6 Using a Redundant Management Module

Overview 175

How management module redundancy works 175
Management module redundancy overview 175
Management module switchover 176
Switchover implications 177

Management module redundancy configuration 178
Changing the default active chassis slot 178

Managing management module redundancy 179
File synchronization between active and standby management modules 179
Manually switching over to the standby management module 181
Rebooting the active and standby management modules 181

Monitoring management module redundancy 182
Determining management module status 182
Monitoring the status change of a module 183
Displaying temperature information 184

Displaying switchover information 184

Flash memory and PCMCIA flash card file management commands 185

Verifying available flash space on the management module before an image is copied 186
Management focus 187
Flash memory file system 188
PCMCIA flash card file system 189
Wildcards 190
Formatting a flash card 190
Determining the current management focus 191
Switching the management focus 191
Displaying a directory of the files 192
Displaying the contents of a file 194
Displaying the hexadecimal output of a file 195
Creating a subdirectory 195
Removing a subdirectory 197
Renaming a file 198
Changing the read-write attribute of a file 198
Deleting a file 199
Recovering (“undeleting”) a file 200
Appending a file to another file 200

Copying files using the copy command 201
Copying files using the cp command 206
Loading the software 206
Saving configuration changes 208
File management messages 209

Chapter 7 NetIron MLX Link Aggregation

Overview 211
LAG formation rules 211
LAG load sharing 214
Hash based load sharing 214
Per packet server LAG load sharing 217
Configuring a LAG 217
Creating a Link Aggregation Group (LAG) using the LAG ID option 218
Deploying a LAG 222
Commands available under LAG once it is deployed 223
Configuring ACL-based mirroring 223
Disabling ports within a LAG 224
Enabling ports within a LAG 224
Adding a Port to Currently Deployed LAG 224
Deleting a Port from a Currently Deployed LAG 224
Monitoring an individual LAG port 225
Assigning a name to a port within a LAG 226
Enabling sFlow forwarding on a port in a LAG 226
Setting the sFlow sampling rate for a port in a LAG 226
Configuring a dynamic LAG within a VRF 227
Displaying LAG information 227
Displaying LAG statistics 232

Chapter 8 VLANs

Overview 235
Tagged, untagged, and dual mode ports 236
Protocol-based VLANs 237
VLAN configuration rules 238
VLAN ID range 238
Tagged VLANs 238
VLAN hierarchy 238
Multiple VLAN membership rules 239
Dual-mode default VLAN 239
Layer 2 control protocols on VLANs 240
Virtual interfaces and CPU protection co-existence on VLANs 241
Configuring port-based VLANs 241
Strictly or explicitly tagging a port 242
Assigning or changing a VLAN priority 242
Assigning a different ID to the default VLAN 243

Configuring protocol-based VLANs 243
Configuring virtual routing interfaces 244
Integrated Switch Routing (ISR) 244
VLAN groups 246
Configuring a VLAN group 247
Configuring super aggregated VLANs 248
Configuring aggregated VLANs 250
Complete CLI examples 252
Configuring 802.1q-in-q tagging 255
Configuration rules 256
Enabling 802.1Q-in-Q tagging 256
Example configuration 257
Configuring 802.1q tag-type translation 257
Configuration rules 259
Miscellaneous VLAN features 260
Allocating memory for more VLANs or virtual routing interfaces 260
Configuring uplink ports within a port-based VLAN 261
Configuring control protocols in VLANs 261
Hardware flooding for layer 2 multicast and broadcast packets 261
Unknown unicast flooding on VLAN ports 262
Configuring VLAN CPU protection 262
Command changes to support 8x10G modules 263
Deprecated commands 263
Existing display command: 265
Extended VLAN counters for 8x10G modules 265
Configuring extended VLAN counters 266
Enabling accounting on per-slot basis 266
Enabling accounting on switched or routed packets 267
Displaying VLAN counters 267
Clearing extended VLAN counters 269
Clearing counters for all VLANs 269
Clearing counters for a specific VLAN 269
Clearing VLAN and port counters 270
Clearing VLAN counters on a port with a specific priority 270
Clearing extended counters statistics on a port 270
Clearing extended counters statistics on specific slot 271
IP interface commands 271
Displaying IP interface counters 271
Displaying IP virtual interface counters 271
Displaying detailed IP virtual interface counters 272
Clearing IP interface counters 273
Clearing IP virtual interface counters 273

Transparent VLAN flooding 274

Displaying VLAN information 275
Displaying VLAN information 275
Displaying VLAN information for specific ports 277
Displaying VLAN status and port types. 277
Displaying VLAN group information 278

Configuring multi-port static MAC address 279
Limitations. 279
Error messages 279

Displaying multi-port static MAC address information 280
Displaying running configuration 281
Displaying changes in the MAC table 281

SA and DA learning and aging 281

MP switchover and hitless upgrade. 281

Flooding features 282

Chapter 9 Configuring Quality of Service for the NetIron MLX

Ingress Traffic processing through a router. 284
Recognizing inbound packet priorities and mapping to internal priority. 285

Creating an Ingress decode policy map 286

Forcing or merging the priority of a packet 286

Forcing or merging the drop precedence of a packet. 287

Egress Traffic processing exiting a router 288

Creating an egress encode policy map 288

Default QoS mappings 288

Protocol Packet Prioritization 293

Configuring QoS. 294
Configuring Ingress QoS procedures 294
Configuring Egress QoS procedures 295
Configuring QoS procedures applicable to Ingress and Egress 295
Configuring Ingress decode policy maps 295
Binding Ingress decode policy maps 301
Configuring a force priority 305
Configuring Egress encode policy maps. 307
Binding an Egress encode EXP policy map 311
Enabling a port to use the DEI bit for Ingress and Egress processing. 316
Specifying the trust level and enabling marking 316
Packet mapping commands 318
Configuring support for super aggregate VLANs 320
Configuring port-level QoS commands on LAG ports 320

Displaying QoS information 321
Displaying QoS configuration information 321

Displaying QoS packet and byte counters 324
Weighted Random Early Discard (WRED) 326
Configuring packet drop priority using WRED 328
Displaying the WRED configuration 334
Scheduling traffic for forwarding 334
Configuring traffic scheduling. 335
Egress port and priority based rate shaping. 337
Multicast queue size, flow control and rate shaping. 338
Traffic manager statistics display 340
Displaying all traffic manager statistics for a router 341
Displaying traffic manager statistics for a port group. 341
Displaying traffic manager statistics for an interface module 342
Displaying traffic manager statistics for NI-MLX-10Gx8-M and NI-MLX-10Gx8-D modules. 344
Displaying traffic manager statistics for the 4x10G module 344
Displaying traffic manager statistics for the 24x1G module 345
Clearing traffic manager statistics 346
New network processor counters displayed for packets to and from traffic manager 347
QoS for NI-MLX-1Gx48-T modules 348
Limitations on TM ports. 348
Configuring priority queues from 8 to 4 348
QoS commands affected by priority queues 349
Priority-based rate shaping 349
Weighted Random Early Discard (WRED). 349
Weighted-based scheduling and mixed strict priority 350
Error messages for CPU copy queue and traffic manager statistics 350
CPU copy queue 350
Traffic manager statistics 350

Chapter 10 Configuring Traffic Policing for the NetIron MLX

Traffic policing on the PowerConnect 353
Layer 2 ACL-based rate limiting 366
Configuration rules and notes. 367
Editing a Layer 2 ACL Table 367
Define rate limiting parameters 367
Binding Layer 2 ACL-based rate limiting policy to a port 367
Specifying rate limiting parameters without a policy map . . .368
Display accounting. 368
Rate limiting protocol traffic using Layer 2 inbound ACLs . . .368
Example of Layer 2 ACL to rate limit broadcast traffic 369
Rate limiting ARP packets. 369
Configuring rate limiting of ARP packets 370
Displaying statistics for ARP rate limiting. 370

Clearing Statistics for ARP Rate Limiting 370

Chapter 11 Configuring Spanning Tree Protocol

Overview 371

IEEE 802.1D Spanning Tree Protocol (STP) 371
Enabling or disabling STP 371
Default STP bridge and port parameters 373
Changing STP bridge parameters 374
Changing STP port parameters 374
Root Guard 375
BPDU Guard 377
Displaying STP information 380

IEEE Single Spanning Tree (SSTP) 386
SSTP defaults 386
Displaying SSTP information 387

SuperSpan™ 388
Customer ID 388
BPDU forwarding 389
Preforwarding state 389
Combining single STP and multiple spanning trees 390
Configuring SuperSpan 394
Displaying SuperSpan information 396

PVST or PVST+ compatibility 396
Overview of PVST and PVST+ 397
VLAN Tags and dual mode 397
Enabling PVST+ support 398
Displaying PVST+ support information 399
Configuration examples 399

802.1s Multiple Spanning Tree Protocol 401
Configuring STP under an ESI VLAN 412

Chapter 12 Configuring Rapid Spanning Tree Protocol

Bridges and bridge port roles 413
Assignment of port roles 414
Ports on Switch 1 415
Ports on Switch 2 415
Ports on Switch 3 415
Ports Switch 4 416

Edge ports and Edge port roles 416

Point-to-point ports 417

Bridge port states 417

Edge port and non-Edge port states 418

Changes to port roles and states 418

State machines 418
Handshake mechanisms 419

Convergence in a simple topology 430
Convergence at start up 431
Convergence after a link failure 433
Convergence at link restoration 434
Convergence in a complex RSTP topology. 435
Propagation of topology change 438
Compatibility of RSTP with 802.1D 441
Configuring RSTP parameters 442
Enabling or disabling RSTP in a port-based VLAN 442
Enabling or disabling RSTP on a single spanning tree 443
Disabling or enabling RSTP on a port. 443
Changing RSTP bridge parameters. 443
Changing port parameters 444
Displaying RSTP information 445
Configuring RSTP under an ESI VLAN 449

Chapter 13 Metro Ring Protocol

Overview 451
Metro Ring Protocol (MRP) 451
MRP rings without shared interfaces (MRP Phase 1) 453
Ring initialization. 454
How ring breaks are detected and healed 457
Master VLANs and member VLANs in a topology group. 461
Configuring MRP 462
Adding an MRP ring to a vlan 463
Changing the hello and preforwarding times. 463
Changing the scale timer. 464
MRP Phase 2. 465
Ring interface ownership. 467
Ring interface IDs and types. 468
Selection of the master node for a ring 469
RHP processing in rings with shared interfaces 471
How ring breaks are detected and healed between shared interfaces. 472
Normal flow 473
Flow when a link breaks 474
Configuring MRP with shared interfaces 475
Tuning MRP timers 476
Flushing the mac table following an MRP event 476
Hello time 476
Preforwarding time. 476
Setting hello and preforwarding timers appropriately. 477
Effect of the scale timer 478
Using MRP diagnostics 479
Enabling MRP diagnostics. 479

Displaying MRP diagnostics 479
Displaying MRP information 480
Displaying topology group information 480
Displaying ring information 480
MRP CLI example 482
Commands on Switch A (master node) 483
Commands on Switch B 483
Commands on Switch C. 484
Commands on Switch D 484
Configuring MRP under an ESI VLAN 485

Chapter 14 Ethernet Ring Protection Protocol

Ethernet Ring Protection. 487
Ethernet Ring Protection components 488
Initializing a new ERN 492
Signal fail 496
Manual switch 498
Forced switch. 500
Double Forced Switch 503
Dual-end blocking 503
Non-revertive mode. 503
Interconnected rings 503
FBD flush optimization 505
Configuring ERP. 505
Sample configuration 505
Configuring ERP with IEEE 802.1ag. 506
ERP commands 507
Assigning ERP IDs 507
Naming an Ethernet Ring Node 507
Configuring the default MAC ID. 507
Enabling the ERP configuration 508
Configuring interfaces 508
Assigning the RPL owner role and setting the RPL 508
Enabling sub-rings for multi-ring and ladder topologies 509
Configuring non-revertive mode 509
Configuring and clearing a forced switch 509
Configuring and clearing a manual switch 509
Configuring dual-end blocking 510
Configuring the guard timer 510
Configuring and clearing the wait to restore timer 511
Testing the WTR timer 511
Configuring and clearing the WTB timer 511
Configuring a hold-off timer 512
Configuring IEEE 802.1ag support 512
Setting the ITU-T G.8032 version number 512

Viewing ERP operational status and clearing ERP statistics. 513
Viewing ERP operational status and statistics. 513
Clearing ERP statistics. 514

Chapter 15 Virtual Switch Redundancy Protocol (VSRP)

Layer 2 redundancy 517
Master election and failover 517
VSRP failover 517
VSRP priority calculation 518
MAC address failover on VSRP-aware devices. 521
Configuring basic VSRP parameters 522
Configuring optional VSRP parameters 522
VSRP 2 524
Configuring VSRP 2 527
Displaying VSRP 2 527
Displaying VSRP information 534
Displaying VRID information 534
Displaying the active interfaces for a VRID 537
VSRP fast start 537
VSRP slow start 539
VSRP and Foundry MRP signaling 540

Chapter 16 Topology Groups

Overview 543
Master VLAN and member VLANs 543
Master VLANs and customer VLANs in Foundry MRP. 544
Control ports and free ports 544
Configuration considerations 544
Configuring a topology group 545
Adding VPLS VLANs to topology groups 546
Topology group support within an ESI 547
Displaying topology group information 547
Displaying topology group information on a NetIron MLX series router 547

Chapter 17 Configuring VRRP and VRRP-E

Overview 551
Overview of VRRP 552
Standard VRRP. 552
Enhancements to VRRP. 554
Configuring unique virtual MAC addresses per VRID 555
Overview of VRRP-E. 558

ARP behavior with VRRP-E 559
Comparison of VRRP and VRRP-E 560
VRRP and VRRP-E parameters 561
Configuring parameters specific to VRRP 564
Configuring the VRRP version 564
Configuring the Owner for IPv4 564
Configuring the Owner for IPv6 565
Configuring a Backup for IPv4 565
Configuring a Backup for IPv6 566
Configuration rules and feature limitations for VRRP 567
Configuring parameters specific to VRRP-E 567
Configuring IPv4 VRRP-E 567
Configuring IPv6 VRRP-E 568
Configuration rules and feature limitations for VRRP-E 568
Configuring additional VRRP and VRRP-E parameters 569
Authentication type 569
Suppressingf RIP advertisements on backup routers for the backup up interface 570
Hello interval 570
Dead interval 571
Backup hello message state and interval 571
Track port 572
Track priority 572
Backup preempt 573
Master router abdication and reinstatement 573
VRRP-extended slow start 574
VRRP-extended scale timer 574
Displaying VRRP and VRRP-E information for IPv4 575
Displaying summary information 575
Displaying detailed information 577
Displaying statistics 580
Displaying VRRP and VRRP-E information for IPv6 581
Displaying summary information 581
Displaying detailed information 582
Displaying statistics 582
Displaying configuration information for VRRP and VRRP-E 583
Clearing VRRP or VRRP-E statistics 583
Configuration examples 584
VRRP example for IPv4 584
VRRP example for IPv6 585
VRRP-E example for IPv4 587
VRRP-E example for IPv6 588
VRRP-E Extension for Server Virtualization 590
VRRP-E Extension for server virtualization configuration example 591

Packets from the local subnet of the virtual IP address 591
IPv4 VRF support 591
Configuration considerations 592

Chapter 18 Multi-Chassis Trunking (MCT)

About Multi-Chassis Trunk (MCT) 593
How Multi-Chassis Trunking works 594
Configuring MCT 600
Optional cluster operation features 615
Port loop detection 619
MCT failover scenarios 620
Show commands 621
Syslogs and debugging 622
Clear MAC commands 625
MCT configuration examples 627

Multi-Chassis Trunk (MCT) for VRRP or VRRP-E 644
One MCT switch is the VRRP or VRRP-E master router and the other MCT switch is VRRP or VRRP-E backup router 644

Chapter 19 Configuring IP

Overview 649

The IP packet flow 650
ARP cache table 651
Static ARP table 652
IP route table 652
IP forwarding cache 653
IP packet queuing 654

Basic IP parameters and defaults 654
When parameter changes take effect 654
IP global parameters 654
IP interface parameters 658

GRE IP tunnel 659
Considerations in implementing this feature 659
GRE MTU enhancements 660
Configuring a GRE IP Tunnel 660

Multicast over GRE tunnel 667
Configuring PIM GRE tunnel 667
Configuring PIM GRE tunnel using the strict RPF check 668

Tunnel statistics for a GRE tunnel or IPv6 manual tunnel 668
Reload behavior and the source-ingress CAM partition 669
Operational notes 669
Enabling IP tunnel or manual IPv6 statistics 671

Restart global timers 672
Configuring the graceful-restart max-hold-timer 673
Graceful-restart protocols-converge-timer 674

Configuring IP parameters 674
Configuring IP addresses. 674
Enabling hardware forwarding of IP option packets based on L3 destination 677
Configuring domain name server (DNS) resolver 679
Using Telnet and Secure Shell 681
Changing the encapsulation type for IP packets 681
Setting the maximum frame size globally 681
Changing the MTU 682
Changing the router ID. 684
Recalculating the router ID 685
Specifying a single source interface for Telnet, SSH, SNTP, TFTP, TACACS/TACACS+, or RADIUS packets. 685
Configuring an interface as the source for Syslog packets 686
Configuring ARP parameters 686
How ARP works. 686
Rate limiting ARP packets 687
Changing the ARP aging period. 688
Enabling proxy ARP 689
Enabling local proxy ARP 689
Disabling gratuitous ARP requests for local proxy ARP 690
Creating static ARP entries 690
Changing the ARP timer. 691
Changing the ARP pending retry timer 691
Dynamic ARP inspection. 691
ARP poisoning 692
How DAI works 692
Configuring DAI. 693
Displaying ARP inspection information. 696
Clearing ARP inspection counters. 698
DHCP snooping 699
How DHCP snooping works 699
System reboot and the binding database 700
Configuring DHCP snooping 700
Clearing the DHCP binding database 701
DHCP option 82 insertion. 701
Displaying DHCP snooping status and ports 702
Displaying DAI binding entries. 703
Displaying DHCP snooping statistics counters 703
Clearing DHCP snooping counters 705
DHCP snooping configuration example 705
IP source guard 706
Enabling IP source guard. 706
Enabling IP source inspection on a VLAN. 707
Displaying IP source inspection status and ports 707
IP source guard CAM. 708
Configuring IP source guard CAM partition 708
Displaying IP source guard CAM partition 708
Configuring forwarding parameters 709

Configuring the maximum ICMP error message rate 711
Disabling ICMP messages 712
Disabling ICMP redirect messages 714
Configuring static routes 714
Static route configuration 725
Naming a static IP route 728
Configuring a default network route 730
Configuring IP load sharing 731
Configuring IRDP 738
Configuring UDP broadcast and IP helper parameters 740
Configuring BootP or DHCP forwarding parameters 742
Filtering Martian addresses 744
Adding, deleting or modifying Martian addresses 745
IPv6 Over IPv4 tunnels in hardware 746
Configuring a IPv6 IP tunnel 746
Configuring a manual IPv6 tunnel 747
Configuring an automatic 6to4 tunnel 747
Displaying IPv6 tunneling information 752
Displaying IP information 754
Displaying IP interface information 756
Displaying interface name in Syslog 758
Displaying ARP entries 759
Displaying the forwarding cache 761
Dual Active Console 762
Displaying the IP route table 763
Clearing IP routes 767
Displaying IP traffic statistics 768
Displaying GRE tunnel information 770
Displaying GRE and manual IPv6 tunnel statistics 770
Displaying martian addressing information 774

Chapter 20 Layer 2 Access Control Lists

Configuration rules and notes 775
General considerations 775
Configuration considerations for VPLS, VLL, and VLL-Local endpoints 776
Types of Layer-2 ACLs 776
Creating a numbered Layer-2 ACL table 777
Filtering and priority manipulation based on 802.1p priority 778
Inserting and deleting Layer-2 ACL clauses 779
Increasing the maximum number of clauses per Layer-2 ACL table 779
Binding a numbered Layer-2 ACL table to an interface 779
Filtering by MAC address 779
Filtering broadcast traffic 780
Using the priority option 780
Using the priority force option 780
Using the priority mapping option 780

Creating a named Layer-2 ACL table 780
Binding a named Layer-2 ACL table to an interface 781
ACL accounting 781
. . . . . Enabling and disabling ACL accounting on
NetIron MLX device 781
Displaying Layer-2 ACLs 782
Displaying Layer-2 ACL statistics on NetIron MLX device 782
Configuring ACL Deny Logging for Layer-2
inbound ACLs 782

Chapter 21 Access Control List

How the PowerConnect processes ACLs 785
General configuration guidelines 786
Configuration considerations for IPv4 outbound
ACLs on VPLS, VLL, and VLL-Local endpoints 786
Disabling outbound ACLs for switching traffic 787
Globally enabling outbound ACLS for switching traffic 787
Enabling outbound ACLS for switching traffic per port 787
Default ACL action. 788
Types of IP ACLs. 788
ACL IDs and entries. 788
Enabling support for additional ACL statements 789
Configuring numbered and named ACLs. 789
Configuring standard numbered ACLs 789
Configuring extended numbered ACLs 791
Configuring standard or extended named ACLs 800
Displaying ACL definitions 802
Modifying ACLs 803
Adding or deleting a comment 805
Deleting ACL entries 807
Deleting entries from numbered ACLs 807
Deleting entries from named ACLs 808
Applying ACLs to interfaces 808
Reapplying modified ACLs 808
Applying ACLs to a virtual routing interface 809
Enabling ACL duplication check 809
Enabling ACL conflict check 810
Enabling ACL filtering of fragmented or
non-fragmented packets. 810
Configuring the conservative ACL fragment mode 811
ACL filtering for traffic switched within a virtual
routing interface 816
Filtering and priority manipulation based on
802.1p priority. 817
Example using the priority option (IPv4). 817

Example using the priority force option 818
Example using the priority mapping option 818
ICMP filtering for extended ACLs 818
Binding IPv4 inbound ACLs to a management port 821
IP receive ACLs 821
Configuration guidelines for IP receive ACLs 822
Configuring rACLs. 822
Displaying accounting information for rACL statistics 824
Matching on TCP header flags for IPv4 ACLs 824
ACL deny logging 825
ACL accounting 828
Displaying accounting statistics for all ACLs 829

Chapter 22 Policy-Based Routing

Overview 833
Configuration considerations 833
Configuring a PBR policy. 834
Configure the route map 834
Enabling PBR 837
Configuration examples 838
Basic example 838
Setting the next hop. 838
Setting the output interface to the null interface 840
Selectively applying normal routing to packets. 841
LAG formation. 841

Chapter 23 Configuring RIP

Overview 843
RIP parameters and defaults 844
RIP global parameters 844
RIP interface parameters 844
Configuring RIP parameters 845
Enabling RIP 845
Configuring metric parameters 846
Changing the administrative distance 846
Configuring redistribution 847
Configuring route learning and advertising parameters 848
Changing the route loop prevention method 849
Suppressing RIP route advertisement on a VRRP or VRRPE backup interface 850
Using prefix lists and route maps as route filters 850
Setting RIP timers 851
Displaying RIP Information 852

Chapter 24 Configuring OSPF Version 2

Overview 857
OSPF point-to-point links 859
Designated routers in multi-access networks 859
Designated router election in multi-access networks 860
OSPF RFC 1583 and 2328 compliance 861
Reduction of equivalent AS external LSAs 861
Support for OSPF RFC 2328 Appendix E 863

OSPF graceful restart 864
Hitless upgrade support for OSPF graceful restart 864
OSPF Stub Router Advertisement 865
OSPF Shortest Path First throttling 865
IETF RFC and internet draft support 866
Dynamic OSPF activation and configuration 867

OSPF VRF-Lite for customer-edge routers 867

Configuring OSPF 867
Configuration rules 868
OSPF parameters 868
Enable OSPF on the router 869
Assign OSPF areas 870
Assign a totally stubby area 870
Assigning an area range (optional) 874
Assigning interfaces to an area 874
Modify interface defaults 874
Change the timer for OSPF authentication changes 877
Block flooding of outbound LSAs on specific OSPF interfaces 878
Assign virtual links 879
Modify virtual link parameters 881
Changing the reference bandwidth for the cost on OSPF interfaces 883
Define redistribution filters 884
Modify default metric for redistribution 886
Enable route redistribution 886
Disable or re-enable load sharing 888
Configure external route summarization 889
Configure default route origination 890
Supported match and set conditions 891

OSPF non-stop routing 892

Synchronization of critical OSPF elements 892
Link state database synchronization 892
Neighbor router synchronization 893
Interface synchronization 893

BFD with OSPF NSR 894

Standby module operations 894
Neighbor database 894
LSA database 894

Enabling and disabling NSR 895

Limitations of NSR 895
Adding additional parameters 895
Disabling configuration 896
OSPF distribute list 897
Modify SPF timers 900
Modify redistribution metric type 900
Modify administrative distance. 901
Configure OSPF group Link State Advertisement (LSA) pacing 902
Modify OSPF traps generated 902
Modify OSPF standard compliance setting 903
Modify exit overflow interval 903
Specify types of OSPF Syslog messages to log 904
Configuring an OSPF network type 904
Configuring OSPF Graceful Restart. 905
Configuring OSPF router advertisement. 907
Configuring OSPF shortest path first throttling 909
Displaying OSPF information 910
Displaying general OSPF configuration information 911
Displaying CPU utilization and other OSPF tasks. 912
Displaying OSPF area information 913
Displaying OSPF neighbor information. 914
Displaying OSPF interface information. 916
Displaying OSPF interface brief information 918
Displaying OSPF route information. 919
Displaying OSPF database information 921
Displaying OSPF external link state information 923
Displaying OSPF database-summary information 924
Displaying OSPF database link state information 924
Displaying OSPF ABR and ASBR information 925
Displaying OSPF trap status 926
Viewing Configured OSPF point-to-point links 926
Displaying OSPF virtual neighbor and link information. 928
Clearing OSPF neighbors. 930
Displaying an OSPF Graceful Restart information. 930
Displaying OSPF Router Advertisement information 931
Clearing OSPF information 931
Clearing OSPF neighbors. 932
Disabling and re-enabling the OSPF process. 932
Clearing OSPF routes. 932

Chapter 25 Configuring IS-IS (IPv4)

Overview 933
Relationship to IP route table 934
Intermediate systems and end systems. 934
Domain and areas 935
Level-1 routing and Level-2 routing 935
Neighbors and adjacencies. 936
Designated IS. 936
Three-way handshake for point-to-point adjacencies 938

IS-IS CLI levels 938
Global configuration level 938
Address family configuration level 939
Interface level. 939
Enabling IS-IS globally 939
Globally configuring IS-IS on a device 940
Setting the overload bit 941
Configuring authentication 942
Changing the IS-IS level globally 945
Disabling or re-enabling display of hostname 945
Changing the Sequence Numbers PDU interval 946
Changing the maximum LSP lifetime 946
Changing the LSP refresh interval 947
Changing the LSP generation interval 947
Changing the LSP interval and retransmit interval 947
Changing the SPF timer. 948
Configuring the IS-IS PSPF exponential back-off feature 948
Configuring the IS-IS flooding mechanism 949
Globally disabling or re-enabling hello padding. 949
Logging adjacency changes 950
Logging invalid LSP packets received. 950
Disabling partial SPF optimizations 950
Disabling incremental SPF optimizations. 951
Configuring IPv4 address family route parameters 951
Changing the metric style 951
Changing the maximum number of load sharing paths 952
Enabling advertisement of a default route 952
Changing the administrative distance for IPv4 IS-IS 953
Configuring summary addresses 954
Redistributing routes into IPv4 IS-IS 954
Changing the default redistribution metric 955
Redistributing static IPv4 routes into IPv4 IS-IS. 955
Redistributing directly connected routes into IPv4 IS-IS 956
Redistributing RIP routes into IPv4 IS-IS 956
Redistributing OSPF routes into IPv4 IS-IS 956
Redistributing BGP4+ routes into IPv4 IS-IS 957
Redistributing IPv4 IS-IS routes within IPv4 IS-IS 957
Configuring IS-IS point-to-point over Ethernet. 958
PowerConnect IS-IS Router A configuration 958
PowerConnect IS-IS Router B configuration. 958
Displaying IS-IS point-to-point configuration 959
Configuring IS-IS over a GRE IP tunnel 959
Configuration considerations 959
Configuring IS-IS over a GRE IP tunnel 960
Displaying IS-IS over GRE IP tunnel 961
IS-IS Non-Stop Routing 963
Overview 963
Limitations 963
Enabling and disabling IS-IS NSR. 964

Displaying the IS-IS NSR status 964
Configuring ISIS properties on an interface 968
Disabling and enabling IS-IS on an interface 968
Disabling or re-enabling formation of adjacencies 969
Setting the priority for designated IS election 969
Limiting access to adjacencies with a neighbor 970
Changing the IS-IS level on an interface 970
Disabling and enabling hello padding on an interface 970
Changing the hello interval 971
Changing the hello multiplier 971
Changing the metric added to advertised routes 971
Displaying IPv4 IS-IS information 972
Displaying ISIS general information 972
Displaying the IS-IS configuration in the running-config 976
Displaying the name mappings 976
Displaying neighbor information 977
Displaying IS-IS Syslog messages 979
Displaying interface information 979
Displaying route information 983
Displaying LSP database entries 984
Displaying traffic statistics 988
Displaying error statistics 989
Displaying the IS-IS SPF Log 992
Clearing the IS-IS SPF Log 994
Triggering the router to run SPF 994
Clearing IS-IS information 995
Clearing a specified LSP from IS-IS database 996

Chapter 26 Configuring BGP4 (IPv4)

Overview 997
Overview of BGP4 998
Relationship between the BGP4 route table and the IP route table 999
How BGP4 selects a path for a route 1000
BGP4 message types 1001
Implementation of BGP4 1003
Memory considerations 1004
Grouping of RIB-out peers 1004
BGP4 Restart 1004
BGP4 Peer notification during a management module switchover 1005
BGP4 neighbor local AS 1006
BGP4 null0 routing 1007
Configuring BGP4 1008
Enabling and disabling BGP4 1012
Disabling BGP4 1012

Entering and exiting the address family configuration level 1013
Aggregating routes advertised to BGP4 neighbors.....1013
Configuring the device to always compare Multi-Exit Discriminators.....1014
Disabling or re-enabling comparison of the AS-Path length 1015
Redistributing IBGP routes.....1015
Disabling or re-enabling client-to-client route reflection 1016
Configuring a route reflector.....1016
Enabling or disabling comparison of device IDs.....1016
Configuring confederations 1017
Four-byte Autonomous System Numbers (AS4) 1020
Enabling AS4 numbers 1021
BGP4 AS4 attribute errors 1025
Error logs 1025
Specifying a maximum AS path length 1026
Setting a global maximum AS path limit 1027
Setting a maximum AS path limit for a peer group or neighbor.....1027
BGP4 max-as error messages 1027
Configuring route flap dampening..... 1028
Originating the default route 1029
Changing the default local preference 1029
Changing the default metric used for redistribution..... 1030
Changing the default metric used for route cost 1030
Changing administrative distances 1030
Requiring the first AS to be the neighbor AS..... 1032
Enabling fast external fallover..... 1033
Setting the local AS number..... 1033
Configuring BGP4 multipath load sharing..... 1034
Customizing BGP4 multipath load sharing 1034
Enhancements to BGP4 load sharing 1035
Configuring a static BGP4 network..... 1035
Configuring paths without MEDs as the least favorable 1037
Configuring BGP4 neighbors 1037
Auto shutdown of BGP4 neighbors on initial configuration 1042
Removing route dampening from suppressed routes..... 1043
Encrypting BGP4 MD5 authentication keys..... 1044
Configuring a BGP4 peer group 1046
Peer group parameters 1046

Specifying a list of networks to advertise 1049
Using the IP default route as a valid next-hop for a BGP4 route 1051
Enabling next-hop recursion 1051
Modifying redistribution parameters 1054
Using a table map to set the tag value 1057
Changing the Keep Alive Time and Hold Time 1057
Changing the BGP4 next-hop update timer 1058
Changing the device ID 1058
Adding a loopback interface 1059
Changing the maximum number of paths for BGP4 load sharing 1059
Configuring route reflection parameters 1060
Filtering 1063
Filtering AS-paths 1063
Filtering communities 1065
Defining and applying IP prefix lists 1067
Defining neighbor distribute lists 1068
Defining route maps 1068
Route-map continue clauses for BGP4 routes 1077
Specifying route-map continuation clauses 1078
Dynamic route filter update 1080
Configuring cooperative BGP4 route filtering 1083
Configuring route flap dampening 1085
Generating traps for BGP4 1089
Updating route information and resetting a neighbor session 1090
Clearing traffic counters 1096
Clearing route flap dampening statistics 1097
Removing route flap dampening 1097
Clearing diagnostic buffers 1097
Configuring BGP4 Restart 1098
Configuring BGP4 null0 routing 1099
Generalized TTL Security Mechanism support 1103
Displaying BGP4 information 1104
Displaying summary BGP4 information 1104
Displaying the active BGP4 configuration 1107
Displaying summary neighbor information 1107
Displaying BGP4 neighbor information 1109
Displaying peer group information 1120
Displaying summary route information 1120
Displaying the BGP4 route table 1121
Displaying BGP4 route-attribute entries 1127
Displaying the routes BGP4 has placed in the IP route table 1129
Displaying route flap dampening statistics 1129
Displaying the active route map configuration 1130

Displaying BGP4 restart neighbor information 1131
Displaying AS4 details 1131
Displaying route-map continue clauses 1135

Chapter 27 Configuring IP Multicast Protocols

Overview 1139

Overview of IP multicasting 1140
Multicast terms 1140

Changing global IP multicast parameters 1140
Concurrent support for multicast routing and snooping 1141
Defining the maximum number of DVMRP cache entries 1141
Defining the maximum number of DVMRP routes. 1142
Defining the maximum number of PIM cache entries. 1142
Defining the maximum number of multicast VRF CAM entries 1143
Defining the maximum number of IGMP group addresses 1143
Changing IGMP V1 and V2 parameters 1144

Support for Multicast Multi-VRF 1146
System max parameter changes 1146
show and clear command support 1146

Adding an interface to a multicast group 1147

Multicast non-stop routing 1148
Configuration considerations 1148
Configuring multicast non-stop routing 1148
Displaying the multicast NSR status 1149

Passive Multicast Route Insertion (PMRI) 1150
Configuring PMRI 1151
Displaying hardware-drop 1151

IP multicast boundaries 1151
Configuring multicast boundaries. 1152
Displaying multicast boundaries. 1152

PIM Dense 1153
Initiating PIM multicasts on a network 1153
Pruning a multicast tree 1154
Grafts to a multicast tree 1156
PIM DM versions 1156
Configuring PIM DM 1156
Failover time in a multi-path topology 1160
Modifying the TTL threshold 1160
Configuring a DR priority 1161
Displaying basic PIM Dense configuration information 1161
Displaying all multicast cache entries in a pruned state 1163

PIM Sparse 1163

PIM Sparse device types 1164
RP paths and SPT paths 1165
Configuring PIM Sparse 1165
ACL based RP assignment 1169
Route selection precedence for multicast 1170
Multicast Outgoing Interface (OIF) list optimization 1173
Displaying PIM Sparse configuration information and statistics 1174
Clearing the PIM forwarding cache 1183
Displaying PIM traffic statistics 1183
Clearing the PIM message counters 1184
Displaying PIM counters 1184
Configuring Multicast Source Discovery Protocol (MSDP) 1185
Configuring MSDP mesh groups 1197
Configuring MSDP mesh group 1198
MSDP Anycast RP 1199
PIM Anycast RP 1203
Configuring PIM Anycast RP 1203
DVMRP overview 1205
Initiating DVMRP multicasts on a network 1206
Pruning a multicast tree 1206
Grafts to a multicast tree 1208
Configuring DVMRP 1208
Enabling DVMRP globally and on an interface 1208
Modifying DVMRP global parameters 1209
Modifying DVMRP interface parameters 1211
Displaying DVMRP information 1212
Configuring a static multicast route 1219
Configuring a static multicast route within a VRF 1220
IGMP V3 1221
Source-specific multicast 1231
Configuring PIM SSM group range 1232
Configuring multiple SSM group ranges 1232
IGMPv2 SSM mapping 1234
IP multicast traffic reduction 1236
Configuration requirements 1236
Configuring IP multicast traffic reduction 1237
PIM SM traffic snooping 1239
Multicast traffic reduction per VLAN or VPLS instance 1243
Static IGMP membership 1246
Displaying IP multicast information 1249

Chapter 28 Configuring MBGP

Overview 1255
Configuration considerations 1256
Configuring MBGP 1256

Setting the maximum number of multicast routes supported 1257
Enabling MBGP 1258
Adding MBGP neighbors 1259
Optional configuration tasks 1260

Displaying MBGP information 1263
Displaying summary MBGP information 1264
Displaying the active MBGP configuration 1264
Displaying MBGP neighbors 1266
Displaying MBGP routes 1268
Displaying the IP Multicast Route Table 1268
Displaying MBGP Attribute Entries 1269
Displaying dampened paths 1270
Displaying MBGP filtered routes 1271
Displaying MBGP flap statistics 1272
Displaying MBGP peer groups 1273

Clearing MBGP information 1274
Clearing route flap dampening information 1274
Clearing route flap statistics 1274
Clearing local information 1274
Clearing BGP neighbor information 1274
Clearing BGP routes 1275
Clearing traffic counters 1275
Clearing VPN4 address family 1275
Clearing VPN Routing/Forwarding information 1275

Chapter 29 Configuring Multi-VRF

Overview of Multi-VRF 1277
Benefits and applications of Multi-VRF 1279
Summary 1282

Configuring Multi-VRF 1282
Configuration 1 1284
Configuration 2 1287

Chapter 30 Configuring MPLS Traffic Engineering

Overview 1291

IETF RFC and Internet draft support 1292
MPLS 1292
OSPF 1292
ISIS 1292

How MPLS works 1292
How packets are forwarded through an MPLS domain 1293

Using MPLS in traffic engineering 1296
CSPF calculates a traffic-engineered path 1297
OSPF-TE Link State Advertisements for MPLS interfaces 1298

IS-IS Link State Protocol data units with TE extensions for MPLS interfaces 1299
Traffic engineering database 1300
LSP attributes and requirements used for traffic engineering 1300
How CSPF calculates a traffic-engineered path 1301
How RSVP establishes a signalled LSP 1302
MPLS fast reroute using one-to-one backup 1310
MPLS Fast Reroute using facility backup over a bypass LSP 1311
MPLS over virtual Ethernet interfaces 1315
Configuration considerations before enabling MPLS on a VE interface 1315
Configuring MPLS 1318
Enabling MPLS 1319
RSVP message authentication 1327
Configuring MPLS on a VE interface 1328
RSVP message authentication on an MPLS VE interface 1332
Setting up signalled LSPs 1333
Configuring signalled LSP parameters 1334
Configuring an adaptive LSP 1347
Configuring MPLS Fast Reroute using one-to-one backup 1350
Protecting MPLS LSPs through a bypass LSP 1352
Displaying MPLS and RSVP information 1354
Displaying information about MPLS-enabled interfaces 1354
Displaying MPLS statistics 1355
Displaying MPLS summary information 1360
Displaying the Traffic Engineering database 1361
Displaying a traffic engineering path to a destination 1364
Displaying signalled LSP status information 1366
Displaying path information 1370
Displaying the MPLS routing table 1371
Displaying RSVP information 1373
Displaying information about OSPF-TE LSAs 1382
Displaying information about IS-IS LSPs with TE extensions 1383
Displaying MPLS Fast Reroute information 1383
Displaying MPLS configuration information 1386
MPLS sample configurations 1390
LSP with redundant paths 1390
Example of MPLS Fast Reroute configuration 1392

Chapter 31 Configuring Label Distribution Protocol (LDP)

LDP overview 1411
Configuring LDP on an interface 1412

Configuring an option of FEC type for auto-discovered VPLS peers. 1413
LDP ECMP for transit LSR. 1413
Changing the maximum number of LDP ECMP paths. 1414
MPLS OAM support for LDP ECMP 1414
Display changes to commands for LDP ECMP. 1415
Setting the LDP Hello Interval and Hold Timeout values 1416
Setting the LDP Hello interval values 1417
Setting the LDP hold time sent to adjacent LSRs 1418
Determining the LDP Hold Time on an MPLS interface 1419
LDP message authentication 1420
Resetting LDP neighbors. 1421
LDP over RSVP (for transit LSR only) 1423
Enabling LDP over RSVP 1424
Configuring a targeted peer address 1426
Displaying targeted peer addresses. 1426
TTL propagation for LDP over RSVP packets 1427
Enabling TTL propagation 1427
Class of Service (CoS) treatment for LDP over RSVP 1427
Displaying LDP information 1428
Displaying the LDP version 1428
Displaying information about LDP-created LSPs 1429
Displaying LDP tunnel LSP information 1430
Displaying the contents of the LDP database 1430
Displaying LDP session information 1431
Displaying LDP neighbor connection information 1433
Displaying information about LDP-enabled interfaces 1434
Displaying information about specified LDP-enabled nterface 1435
Displaying the LDP peer information 1436
Display considerations for LDP FEC information 1438
Displaying LDP FEC information 1438
Displaying information for a specified LDP FEC type. 1439
Displaying LDP FEC summary information. 1440
Displaying the LDP FEC VC information 1441
Displaying information for a specified LDP FEC VC 1441
Displaying the LDP packet statistics. 1444
Clearing the LDP packet statistics 1445
Sample LDP configurations 1445
Router R1 1445
Router R2 1446
Router R3 1446
Sample LDP configuration with VLL. 1447
Router R1 1447
Router R2 1448
Router R3 1448

Chapter 32 Configuring MPLS Virtual Leased Line

Overview 1451

How MPLS VLL works 1452
MPLS VLL packet encoding 1453
QoS for VLL traffic 1453
CoS behavior for VLL tagged mode and VLL raw mode . . . 1455
Configuring MPLS VLLs 1460
Creating a VLL 1460
Specifying tagged or raw mode for a VLL 1460
Specifying a VLL peer 1461
Specifying a VLL endpoint 1462
Enabling VLL MTU enforcement (optional) 1466
Specifying a VLL MTU 1466
Generating traps for VLLs 1466
MPLS VLL behavior with other features 1467
sFlow 1467
IFL CAM 1467
Layer 2 ACLs 1468
Displaying MPLS VLL information 1468
Displaying information about MPLS VLLs 1468
Displaying detailed information about MPLS VLLs 1469
Displaying LDP information 1474
Displaying VLL endpoint statistics 1475
Clearing VLL traffic statistics 1476
Sample MPLS VLL configuration 1477
Router R1 1477
Router R2 1478
Router R3 1479
Local VLL 1479
Local VLL configuration examples 1480
Local VLL QoS 1482
CoS behavior for Local VLL 1484
Configuring Local VLL 1485
Displaying Local VLL information 1487
Displaying information about Local VLLs 1488
Displaying Local VLL endpoint statistics 1490
Enabling MPLS Local VLL traps 1491
Disabling Syslog messages for MPLS VLL-local and VLL 1491

Chapter 33 Configuring MPLS Virtual Private LAN Services

Overview 1493
How VPLS works 1494
Configuring VPLS instances 1496
Creating a VPLS instance 1496
Specifying VPLS peers 1498
Setting the VPLS VC mode 1499
QoS for VPLS traffic 1502
Specifying an LSP to reach a peer within a VPLS 1503
LSP load balancing for VPLS traffic 1504

Specifying the endpoint of a VPLS instance 1505
Flooding Layer 2 BPDUs in VPLS. 1510
Specifying the VPLS VC type 1511
Configuring VPLS tagged mode. 1511
VPLS CPU protection 1512
VPLS Broadcast, multicast, and unknown unicast packet limiting 1514
Layer 2 control traffic behavior on VPLS endpoints 1515
802.1x Protocol packets on a VPLS endpoint 1515
Cisco Discovery Protocol packets 1515
Foundry Discovery Protocol packets. 1515
Uni-directional Link Detection packets. 1515
Flooding Layer 2 BPDUs with a VPLS instance 1516
Specifying a VPLS MTU 1516
Configuring VPLS MTU enforcement. 1517
Configuring VPLS local switching 1517
Enabling MPLS VPLS traps 1518
Disabling Syslog messages for MPLS VPLS 1518
Local VPLS 1518
Example Local VPLS configuration 1519
CoS behavior for Local VPLS 1520
Displaying VPLS information. 1522
Display considerations for VPLS information 1522
Displaying VPLS summary information. 1523
Displaying information about VPLS instances 1523
Displaying detailed information about VPLS instances. . . . 1524
Displaying information about a specified VPLS ID or VPLS name. 1528
Displaying VPLS CPU protection configuration status 1531
Displaying information about VPLS instances that are not operational. 1531
Displaying the contents of the VPLS MAC database 1531
Displaying VPLS traffic statistics. 1534
Clearing VPLS traffic statistics 1535
VPLS LDP 1536
Displaying the VPLS peer FSM state with LDP support. . . . 1536
VC type mismatched 1536
MTU mismatched 1537
No remote VC label 1537
LDP session down 1538
No local label resource 1538
MPLS LDP show commands. 1539
Using the show mpls ldp vc x command 1539

Chapter 34 Configuring BGP-Based Auto-Discovery for VPLS

Overview 1541
Terms introduced in this chapter 1541

How BGP-based auto-discovery for VPLS works 1542
About the L2VPN VPLS address family 1542
Feature limitations and configuration notes 1543
Scalability. 1543
Configuring BGP-based auto-discovery for VPLS 1543
Configuring a loopback interface 1544
Configuring BGP4 to support VPLS auto-discovery 1545
Configuring VPLS to support auto-discovery 1547
Enabling VPLS auto-discovery. 1550
Configuring the L2VPN VPLS address family and activating the BGP4 peering session 1551
Clearing the BGP L2VPN route table 1551
Clearing the BGP L2VPN route table and resetting BGP 1551
Clearing the BGP L2VPN route table without resetting the BGP session 1552
Example configuration 1552
Displaying VPLS auto-discovery information 1556
Displaying information about BGP L2VPN VPLS routes 1556
Displaying information about VPLS auto-discovery and load balancing 1571
Displaying information about LDP 1573

Chapter 35 IP over MPLS

Overview 1575
BGP shortcuts 1575
Key algorithms 1576
Examples of next-hop MPLS 1577
LDP route injection 1581
Considerations when using LDP route injection 1582
LDP route injection example 1582
Displaying routes through LSP tunnels. 1583
Using traffic-engineered LSPs within an AS 1584
IS-IS shortcuts over an LSP tunnel 1585
Creating OSPF shortcuts over an LSP tunnel. 1585
IS-IS shortcuts 1586
Overview 1586
Determining the cost of an IS-IS shortcut 1586
Configuration notes 1587
Configuration tasks 1588
Example configurations. 1590
Clearing IS-IS shortcuts 1592
Show command support 1593
ECMP forwarding for IP over MPLS 1596
QoS mapping between IP packets and MPLS 1596

Chapter 36 Configuring BGP or MPLS VPNs

Overview 1597

What is a BGP or MPLS VPN. 1598
IETF RFC and Internet Draft support 1600

BGP or MPLS VPN components and what they do 1600

BGP or MPLS VPN operation 1601
Creating routes in a BGP or MPLS VPN 1602
Routing a packet through a BGP or MPLS VPN 1602

Configuring BGP VPNs on a PE. 1603
Defining a VRF routing instance 1604
Configuring MPLS forwarding 1604
Assigning a Route Distinguisher to a VRF 1605
Defining IPv4 or IPv6 address families of a VRF 1605
Defining automatic route filtering. 1606
Assigning a VRF routing instance to an interface 1606
Assigning a VRF routing instance to a LAG interface. 1606
Setting up cooperative route filtering. 1607
Importing and exporting route maps 1608
Defining an extended community for use with a route map 1608
Creating a VPNv4 route reflector 1609
Configuring BGP VRF load sharing 1610
ECMP forwarding for IP VPN 1610
Configuring autonomous system number override 1611
Configuring a PE to allow routes with its AS number. 1611
Setting up LSPs per VRF 1612
Configuring OSPF sham links 1612
Adding a static ARP entry for a VRF 1615
Configuring IP TTL to MPLS TTL propagation in an IPVPN. 1616
Configuring a static route within the VRF context 1616
Configuring an IP Static interface route across VRFs 1617
Configuring a backup Virtual Router for VRF using VRRPE. 1617
Ping and Traceroute for layer-3 VPNs 1618
Generating traps for VRFs 1618

Displaying BGP or MPLS VPNv4 information. 1619
Displaying VPNv4 route information. 1620
Displaying VPNv4 route information for a specified IP address 1622
Displaying VPNv4 attribute entries information. 1623
Displaying VPNv4 dampened paths information 1624
Displaying VPNv4 filtered routes information 1624
Displaying VPNv4 Flap statistics information 1624
Displaying VPNv4 route distinguisher information 1625
Displaying VPNv4 neighbor information. 1626
Displaying advertised routes for a specified VPNv4 neighbor 1633

Displaying attribute entries for a specified VPNv4 neighbor 1633
Displaying Flap statistics for a specified VPNv4 neighbor by IP address 1635
Displaying received ORFs information for a specified VPNv4 neighbor 1636
Displaying a specified neighbor VPNv4 routes 1636
Displaying routes summary for a specified VPNv4 neighbor 1638
Displaying summary route information 1640
Displaying the VPNv4 route table 1641
Displaying the best VPNv4 routes. 1642
Displaying best VPNv4 routes that are not in the IP route table 1643
Displaying VPNv4 routes with unreachable destinations . . 1643
Displaying information for a specific VPNv4 route. 1644
Displaying VPNv4 route details 1644
Displaying BGP VPNv4 MPLS tag information 1645
Displaying BGP or MPLS VRF information 1646
Displaying VRF route information 1646
Displaying VRF route information for a specified IP address. 1648
Displaying attribute entries information for a specified VRF 1649
Displaying dampened paths information for a specified VRF 1650
Displaying filtered routes information for a specified VRF 1651
Displaying Flap statistics information for a specified VRF 1651
Displaying BGP neighbor information for a specified VRF 1652
Displaying advertised routes for a specified VRF neighbor 1659
Displaying neighbor attribute entries for a specified VRF 1659
Displaying flap statistics for a specified VRF neighbor by IP address 1661
Displaying received ORF information for a specified VRF neighbor 1661
Displaying received routes for a specified VRF neighbor 1662
Displaying a specified VRF neighbor routes. 1662
Displaying VPNv4 routes summary for a specified VRF neighbor 1664
Displaying summary route information for a specified VRF 1666
Displaying a VRF's BGP4 route table 1667
Displaying additional BGP or MPLS VPN information 1672
Displaying VRF information 1673
Displaying IP network information for a VRF 1673

Displaying the IP route table for a specified VRF 1674
Displaying ARP VRF information 1675
Displaying OSPF information for a VRF 1676
Displaying OSPF area information for a VRF 1676
Displaying OSPF ABR and ASBR information for a VRF 1676
Displaying general OSPF configuration information for a VRF 1677
Displaying OSPF external link state information for a VRF 1678
Displaying OSPF link state information for a VRF 1679
Displaying OSPF interface information 1680
Displaying OSPF neighbor information for a VRF 1680
Displaying the routes that have been redistributed into OSPF 1680
Displaying OSPF route information for a VRF 1681
Displaying OSPF trap status for a VRF 1682
Displaying OSPF virtual links for a VRF 1682
Displaying OSPF virtual neighbor information for a VRF 1682
Displaying IP extcommunity list information 1683
Displaying the IP static route table for a VRF 1683
Displaying the static ARP table for a VRF 1683
Displaying TCP connections for a VRF 1684
Displaying MPLS statistics for a VRF 1684
Displaying IP route information for a VRF 1685
Displaying RIP information for a VRF 1685

BGP or MPLS VPN sample configurations 1686
Basic configuration example for IBGP on the PEs 1686
EBGP for route exchange 1687
Static routes for route exchange 1692
RIP for route exchange 1695
OSPF for route exchange 1700
Cooperative route filtering 1705
Using an IP extcommunity variable with route map 1707
Autonomous system number override 1708
Setting an LSP for each VRF on a PE 1709
OSPF sham links 1710

Chapter 37 IPv6 Addressing

IPv6 addressing overview 1713
IPv6 address types 1714

IPv6 stateless auto-configuration 1716

Chapter 38 Configuring Basic IPv6 Connectivity

Enabling IPv6 routing 1718

Configuring IPv6 on each interface 1718
Configuring a global or site-local IPv6 address 1718
Configuring a link-local IPv6 address 1719
Configuring IPv6 anycast addresses 1720

Configuring the management port for an IPv6 automatic address configuration 1721
IPv6 host support 1721
IPv6 host supported features 1721
IPv6 unsupported features 1721
Restricting SNMP access to an IPv6 node 1722
Specifying an IPv6 SNMP trap receiver 1722
Restricting Telnet access by specifying an IPv6 ACL 1722
Restricting SSH access by specifying an IPv6 ACL 1723
Restricting Web management access by specifying an IPv6 ACL 1723
Restricting SNMP access by specifying an IPv6 ACL 1724
Restricting Web management access to your device to a specific IPv6 host 1724
Specifying an IPv6 Syslog server 1725
Viewing IPv6 SNMP server addresses 1725
Disabling router advertisement and solicitation messages 1726
Configuring IPv4 and IPv6 protocol stacks 1726
Configuring IPv6 Domain Name Server (DNS) resolver 1727
Defining a DNS entry 1727
ECMP load sharing for IPv6 1728
Disabling or re-enabling ECMP load sharing for IPv6 1728
Changing the maximum number of load sharing paths for IPv6 1729
DHCP relay agent for IPv6 1729
Configuring DHCP for IPv6 relay agent 1729
Enabling support for network-based ECMP load sharing for IPv6 1729
Displaying ECMP load-sharing information for IPv6 1730
Configuring IPv6 ICMP 1731
Configuring ICMP rate limiting 1731
Disabling or re-enabling ICMP redirect messages 1732
Disabling ICMP error messages for source-routed IPv6 packets 1732
Enabling ICMP error messages for an unreachable address 1733
Enabling ICMP messages for an unreachable route 1733
Configuring IPv6 neighbor discovery 1733
Neighbor solicitation and advertisement messages 1734
Router advertisement and solicitation messages 1734
Neighbor redirect messages 1735
Setting neighbor solicitation parameters for duplicate address detection 1735
Setting IPv6 router advertisement parameters 1736
Controlling prefixes advertised in IPv6 router advertisement messages 1737
Setting flags in IPv6 router advertisement messages 1737
Configuring reachable time for remote IPv6 nodes 1738

IPv6 source routing security enhancements 1739
Complete filtering of IPv6 source-routed packets 1739
Selective filtering of IPv6 source-routed packets using ACLs 1740
Complete and selective filtering combination and order of application 1741
Configuration examples for complete and selective filtering of source routed packets 1742
Changing the IPv6 MTU 1744
Configuring static neighbor entries 1745
Limiting the number of hops an IPv6 packet can traverse 1746
QoS for IPv6 traffic 1746
Clearing global IPv6 information 1747
Clearing the IPv6 cache 1747
Clearing IPv6 neighbor information 1748
Clearing IPv6 routes from the IPv6 route table 1748
Clearing IPv6 traffic statistics 1749
Deleting IPv6 session flows 1749
Displaying global IPv6 information 1749
Displaying IPv6 cache information 1749
Displaying IPv6 interface information 1750
Displaying interface counters for all ports 1752
Displaying IPv6 neighbor information 1753
Displaying the IPv6 route table 1754
Displaying local IPv6 devices 1756
Displaying IPv6 TCP information 1757
Displaying IPv6 traffic statistics 1760
Displaying IPv6 session flows 1763

Chapter 39 Configuring an IPv6 Prefix List

Configuring an IPv6 prefix list 1767
Displaying prefix list information 1768

Chapter 40 Configuring an IPv6 Access Control List

Configuration considerations for IPv6 outbound ACLs on VPLS, VLL, and VLL-local endpoints 1770
Using IPv6 ACLs as input to other features 1771
Configuring an IPv6 ACL 1771
Example configurations 1771
Default and implicit IPv6 ACL action 1773
ACL syntax 1773
Filtering packets based on DSCP values 1778
Filtering packets based on routing header type 1778
Extended IPv6 ACLs 1779
Configuration considerations for extended IPv6 layer 4 ACL 1779

ACL syntax 1780
CAM partitioning 1784
Applying an IPv6 ACL. 1785
Applying an IPv6 ACL to a VRF 1785
Controlling access to a router 1786
Adding a comment to an IPv6 ACL entry 1787
ACL CAM sharing for inbound IPv6 ACLs 1788
Considerations when implementing this feature. 1788
Configuring ACL CAM sharing for IPv6 ACLs. 1789
Filtering and priority manipulation based on 802.1p priority . . . 1789
Example using the priority force option 1789
Example using the priority mapping option 1790
ACL accounting 1790
Displaying statistics for IPv6 ACL accounting 1791

Chapter 41 Configuring IPv6 Routes

Configuring a static IPv6 route. 1793
Configuring a IPv6 static multicast route 1795

Chapter 42 Configuring RIPng

Configuring RIPng 1797
Enabling RIPng. 1798
Configuring RIPng timers. 1798
Configuring route learning and advertising parameters 1799
Redistributing routes into RIPng. 1801
Controlling distribution of routes through RIPng 1801
Configuring poison reverse parameters. 1802
Clearing RIPng routes from IPv6 route table. 1802
Displaying RIPng information 1803
Displaying RIPng configuration 1803
Displaying RIPng routing table 1804

Chapter 43 Configuring OSPF Version 3

OSPF Version 3 1807
Link-state advertisement types for OSPFv3 1808
Configuring OSPFv3 1808
Enabling OSPFv3 1809
Assigning OSPFv3 areas 1809
Specifying a network type 1811
Configuring virtual links. 1811
Changing the reference bandwidth for the cost on OSPFv3 interfaces 1813
Redistributing routes into OSPFv3 1815
Filtering OSPFv3 routes. 1818
Configuring default route origination 1821

Modifying Shortest Path First timers 1822
Modifying administrative distance 1823
Configuring the OSPFv3 LSA pacing interval 1824
Modifying exit overflow interval. 1824
Modifying external link state database limit 1824
Modifying OSPFv3 interface defaults 1825
Disabling or reenabling event logging 1826
IPsec for OSPFv3 1826
Configuring IPsec for OSPFv3 1827

Displaying OSPFv3 information 1833
General OSPF configuration information 1833
Displaying OSPFv3 area information 1834
Displaying OSPFv3 database information 1835
Displaying IPv6 interface information. 1840
Displaying IPv6 OSPFv3 interface information 1841
Displaying OSPFv3 memory usage 1846
Displaying OSPFv3 neighbor information. 1846
Displaying routes redistributed into OSPFv3 1849
Displaying OSPFv3 route information 1850
Displaying OSPFv3 SPF information 1851
Displaying IPv6 OSPF virtual link information 1854
Displaying OSPFv3 virtual neighbor information 1854
IPsec examples 1855

OSPFv3 clear commands 1864

Chapter 44 Configuring IPv6 IS-IS

IPv6 IS-IS single-topology mode 1867

IS-IS CLI levels 1868
Global configuration level 1869
Address family configuration level 1869
Interface level. 1870

Configuring IPv6 IS-IS 1870
Enabling IPv6 IS-IS globally 1870
Enabling IS-IS and assigning an IPv6 address to an interface 1871

Configuring IPv6 IS-IS single topology 1872

Globally configuring IS-IS on a device 1872

Configuring IPv6 specific address family route parameters . . . 1872
Changing the maximum number of load sharing paths . . . 1873
Enabling advertisement of a default route 1873
Changing the administrative distance for IPv6 IS-IS 1874
Configuring summary prefixes 1875
Redistributing routes into IPv6 IS-IS 1875
Changing the default redistribution metric 1876
Redistributing static IPv6 routes into IPv6 IS-IS. 1876
Redistributing directly connected routes into IPv6 IS-IS 1877
Redistributing RIPng routes into IPv6 IS-IS 1877
Redistributing OSPF version 3 routes into IPv6 IS-IS. 1877

Redistributing BGP4+ routes into IPv6 IS-IS 1878
Redistributing IPv6 IS-IS routes within IPv6 IS-IS 1878
Disabling and re-enabling IPv6 protocol-support consistency checks 1879
Configuring ISIS properties on an interface 1879
Changing the metric added to advertised routes 1879
Displaying IPv6 IS-IS information 1880
Displaying IPv6 IS-IS information 1881
Displaying the IPv6 IS-IS configuration in the running configuration 1882
Displaying IPv6 IS-IS error statistics 1883
Displaying LSP database entries 1884
Displaying the system ID to name mappings 1887
Displaying IPv6 IS-IS interface information 1888
Displaying IPv6 IS-IS memory usage 1890
Displaying IPv6 IS-IS neighbor information 1891
Displaying IPv6 IS-IS redistribution information 1893
Displaying the IPv6 IS-IS route information 1893
Displaying IPv6 IS-IS traffic statistics 1894

Chapter 45 Configuring BGP4+

Address family configuration level 1897
Configuring BGP4+ 1898
Enabling BGP4+ 1898
Configuring BGP4+ neighbors using global or site-local IPv6 addresses 1899
Adding BGP4+ neighbors using link-local addresses 1900
Configuring a BGP4+ peer group 1902
Advertising the default BGP4+ route 1903
Importing routes into BGP4+ 1903
Redistributing prefixes into BGP4+ 1904
Aggregating routes advertised to BGP4 neighbors 1905
Using route maps 1905
Clearing BGP4+ information 1906
Removing route flap dampening 1906
Clearing route flap dampening statistics 1906
Clearing BGP4+ local route information 1907
Clearing BGP4+ neighbor information 1907
Clearing and resetting BGP4+ routes in the IPv6 route table 1910
Clearing traffic counters for all BGP4+ neighbors 1910
Displaying BGP4+ information 1910
Displaying the BGP4+ route table 1911
Displaying BGP4+ route information 1917
Displaying BGP4+ route-attribute entries 1918
Displaying the BGP4+ running configuration 1920
Displaying dampened BGP4+ paths 1920
Displaying filtered-out BGP4+ routes 1921
Displaying route flap dampening statistics 1926
Displaying BGP4+ neighbor information 1927

Displaying BGP4+ peer group configuration information . . 1948
Displaying BGP4+ summary 1949

Chapter 46 Configuring IPv6 Multicast Features

IPv6 PIM Sparse 1951
PIM Sparse router types 1951
RP paths and SPT paths 1952
RFC 3513 and RFC 4007 compliance for IPv6 multicast scope-based forwarding 1952
Configuring PIM Sparse. 1953
IPv6 PIM-Sparse mode 1953
Configuring IPv6 PIM-SM on a virtual routing interface 1954
Enabling IPv6 PIM-SM for a specified VRF. 1954
Configuring BSRs 1954
Route selection precedence for multicast 1958
Enabling Source-specific Multicast. 1962
Configuring a DR priority 1963
Passive Multicast Route Insertion 1964
Displaying PIM Sparse configuration information and statistics 1965
Clearing the IPv6 PIM forwarding cache 1976
Clearing the IPv6 PIM message counters 1976
Updating PIM Sparse forwarding entries with a new RP configuration 1977
Clearing the IPv6 PIM traffic 1977
Setting the maximum number of IPv6 multicast routes supported 1977
Defining the maximum number of IPv6 PIM cache entries 1978
Defining the maximum number of IPv6 multicast VRF CAM entries for all VRFs 1978
Defining the maximum number of IPv6 multicast VRF CAM entries for a specified VRF 1978

PIM Anycast RP 1979
Configuring PIM Anycast RP 1979

Multicast Listener Discovery and source-specific multicast protocols 1981
Enabling MLDv2. 1982
Configuring MLD parameters for default and non-default VRFs 1982
Configuring MLD parameters at the interface level 1985
Displaying MLD information 1986
Clearing IPv6 MLD traffic 1991
Clearing the IPv6 MLD group membership table cache. 1991

IPv6 Multicast Listener Discovery snooping 1992
Configuring IPv6 multicast routing or snooping. 1992
Enabling IPv6 multicast traffic reduction. 1992
PIM-SM traffic snooping 1994

Configuring IPv6 MLD snooping on a per-VLAN basis 1999
Displaying IPv6 multicast information 2002

Chapter 47 Managing a Device Over IPv6

Using the IPv6 copy command 2005
Copying a file to an IPv6 TFTP server 2005
Copying a file from an IPv6 TFTP server 2006
Using the IPv6 ncopy command 2007
Copying a primary or secondary boot image from flash memory to an IPv6 TFTP server 2008
Copying the running or startup configuration to an IPv6 TFTP server 2008
Uploading files from an IPv6 TFTP server 2008
Using the IPv6 ping command 2009
Using the IPv6 traceroute command 2011
Using Telnet 2011
Using the IPv6 Telnet command 2011
Establishing a Telnet session from an IPv6 host 2012
Using Secure Shell 2013

Chapter 48 Configuring Secure Shell and Secure Copy

SSH Version 2 support 2016
Tested SSHv2 clients 2016
Supported features 2016
Configuring SSH 2017
Generating a host key pair 2017
Configuring DSA challenge-response authentication 2018
Setting optional parameters 2020
Disabling 3-DES 2024
Displaying SSH connection information 2024
Ending an SSH connection 2025
Using Secure Copy 2026
Outbound commands: 2026
Inbound commands: 2027

Chapter 49 Configuring Multi-Device Port Authentication

How multi-device port authentication works 2035
RADIUS authentication 2035
Authentication-failure actions 2036
Supported RADIUS attributes 2036
Dynamic VLAN and ACL assignments 2036
Support for authenticating multiple MAC addresses on an interface 2037
Support for multi-device port authentication and 802.1x on the same interface 2037

Configuring multi-device port authentication 2037
Enabling multi-device port authentication 2038
Configuring an authentication method list for 802.1x 2038
Setting RADIUS parameters 2038
Specifying the format of the MAC addresses sent to the RADIUS server 2039
Specifying the authentication-failure action 2040
Defining MAC address filters 2040
Configuring dynamic VLAN assignment 2041
Specifying the VLAN to which a port is moved after the RADIUS-specified VLAN assignment expires 2042
Saving dynamic VLAN assignments to the running configuration file 2043
Clearing authenticated MAC addresses 2043
Disabling aging for authenticated MAC addresses 2043
Specifying the aging time for blocked MAC addresses 2044
Displaying multi-device port authentication information 2044
Displaying authenticated MAC address information 2045
Displaying multi-device port authentication configuration information 2046
Displaying multi-device port authentication information for a specific MAC address or port 2048
Displaying the authenticated MAC addresses 2049
Displaying the non-authenticated MAC addresses 2049

Chapter 50 Using the MAC Port Security Feature

Overview 2051
Local and global resources 2052
Configuring the MAC port security feature 2052
Enabling the MAC port security feature 2052
Setting the maximum number of secure MAC addresses for an interface 2053
Setting the port security age timer 2053
Specifying secure MAC addresses 2054
Autosaving secure MAC addresses to the startup-config file 2054
Specifying the action taken when a security violation occurs 2054
Denying specific MAC addresses 2055
Port security MAC violation limit 2056
Displaying port security information 2057
Displaying port security settings 2057
Displaying the secure MAC addresses on the device 2058
Displaying port security statistics 2059

Chapter 51 Configuring 802.1x Port Security

Overview of 802.1x port security 2061

IETF RFC support 2061
How 802.1x port security works. 2062
Device roles in an 802.1x configuration 2062
Communication between the devices 2063
Controlled and uncontrolled ports 2064
Message exchange during authentication 2065
Authenticating multiple clients connected to the same port. 2066
802.1x port security and sFlow 2068
Configuring 802.1x port security 2068
Configuring an authentication method list for 802.1x. 2069
Setting RADIUS parameters 2069
Configuring dynamic VLAN assignment for 802.1x ports. 2070
Disabling and enabling strict security mode for dynamic filter assignment 2071
Dynamically applying existing ACLs or MAC address filter 2073
Configuring per-user IP ACLs or MAC address filters 2074
Enabling 802.1x port security 2074
Setting the port control 2075
Configuring periodic re-authentication 2076
Re-authenticating a port manually 2077
Setting the quiet period 2077
Setting the interval for retransmission of EAP-request or identity frames 2077
Specifying the number of EAP-request or identity frame retransmissions 2078
Specifying a timeout for retransmission of messages to the Authentication Server 2078
Specifying a timeout for retransmission of EAP-request frames to the client 2078
Initializing 802.1x on a port 2079
Allowing multiple 802.1x clients to authenticate. 2079
Displaying 802.1x information 2080
Displaying 802.1x configuration information. 2080
Displaying 802.1x statistics 2083
Clearing 802.1x statistics 2084
Displaying dynamically assigned VLAN information 2084
Displaying information on MAC address filters and IP ACLs on an interface 2085
Displaying information about the dot1x-mac-sessions on each port. 2086
Sample 802.1x configurations. 2088
Point-to-point configuration. 2088
Hub configuration 2089

Chapter 52 Protecting against Denial of Service Attacks

Protecting against smurf attacks 2091
Avoiding being an intermediary in a smurf attack 2092
Avoiding being a victim in a smurf attack 2092
Protecting against TCP SYN attacks 2093
TCP security enhancement 2094
Displaying statistics from a DoS attack 2096
Clear DoS attack statistics 2096

Chapter 53 Reverse Path Forwarding

Configuration of RPF 2097
Configuration considerations for RPF 2097
Special considerations for configuring RPF with ECMP routes 2098
RPF support for IP over MPLS routes 2098
RPF compatible CAM profiles 2098
Configuring the global RPF command 2099
Enable RPF on individual ports 2100
Configuring a timer interval for IPv6 session logging 2100
Suppressing RPF for packets with specified address prefixes 2101
Excluding packets that match the routers default route 2102
Displaying RPF statistics 2102
Clearing RPF statistics for a specified IPv4 interface 2103
Clearing RPF statistics for all IPv4 interfaces within a router 2103
Clearing RPF statistics for a specified IPv6 interface 2104
Clearing RPF statistics for all IPv6 interfaces within a router 2104
Displaying RPF logging 2104

Chapter 54 Securing SNMP Access

Establishing SNMP community strings 2105
Encryption of SNMP community strings 2105
Adding an SNMP community string 2106
Displaying the SNMP community strings 2107
Using the User-Based Security model 2107
Configuring your NMS 2107
Configuring SNMP version 3 on the PowerConnect 2107
Defining the engine ID 2108
Defining an SNMP group 2109
Defining an SNMP user account 2110
Displaying the engine ID 2111
Displaying SNMP groups 2111
Displaying user information 2112
Interpreting varbinds in report packets 2112
Defining SNMP views 2113

SNMP v3 configuration examples 2114

Chapter 55 Remote Network Monitoring

Basic management 2115
Viewing system information 2115
Viewing configuration information 2115
Viewing port statistics 2116
Viewing STP statistics 2116
Clearing statistics. 2116

RMON support. 2116
Statistics (RMON group 1). 2116
History (RMON group 2). 2119
Alarm (RMON group 3). 2119
Event (RMON group 9). 2120

Chapter 56 Configuring Management VRF

Management VRF overview 2121
Source interface and management VRF compatibility 2122
Supported management applications 2122

Configuring a global management VRF 2125
Configuration notes 2125

Displaying the management VRF information. 2125

Chapter 57 sFlow

Configuration considerations 2129
Source address 2130
Sampling rate 2130

sFlow support for MPLS 2131

sFlow with VPLS local switching. 2131

Configuring and enabling sFlow 2132
Specifying the collector 2132
Changing the polling interval. 2132
Changing the sampling rate 2133
Enabling sFlow forwarding. 2134

ACL-based Inbound sFlow 2135
Configuring ACL-based Inbound sFlow 2136
Displaying sFlow information 2138
Displaying ACL-based sFlow statistics 2139
Clearing sFlow statistics 2139

Chapter 58 Configuring Uni-Directional Link Detection (UDLD)

Configuration considerations 2141

Configuring UDLD 2142
Changing the keepalive interval 2142
Changing the keepalive retries 2142

UDLD for tagged ports 2142
Displaying UDLD information 2143
Displaying information for all ports 2143
Displaying information for a single port 2144
Clearing UDLD statistics 2146

Chapter 59 BiDirectional Forwarding Detection (BFD)

Number of BFD sessions supported 2148
Configuring BFD parameters 2148
Disabling BFD Syslog messages 2149
Displaying BFD information 2149
Displaying BFD information 2149
Clearing BFD neighbor sessions 2153
Configuring BFD for the specified protocol 2153
Configuring BFD for OSPFv2 2153
Configuring BFD for OSPFv3 2154
Configuring BFD for IS-IS 2155
Configuring BFD for BGP4 2156
Displaying BFD for BGP4 2159
.Displaying summary neighbor information 2163
Configuring BFD for RSVP-TE LSPs 2164
BFD session support per-router and per-interface module .2165
BFD session creation 2165
Displaying MPLS BFD information 2167
Displaying BFD application information 2167
Displaying BFD MPLS information 2168
Displaying BFD MPLS detailed information 2169
Displaying MPLS BFD global configuration information 2169

Chapter 60 Operations, Administration, and Maintenance (OAM)

IEEE 802.1ag Connectivity Fault Management (CFM) 2171
Overview 2171
Ethernet OAM capabilities 2172
IEEE 802.1ag purpose 2172
IEEE 802.1ag provides hierarchical network management 2173
Mechanisms of Ethernet IEEE 802.1ag OAM 2173
Fault detection (continuity check message) 2173
Fault verification (Loopback messages) 2174
Fault isolation (Linktrace messages) 2174
Configuring IEEE 802.1ag CFM 2175
Enabling or disabling CFM 2175
Creating a Maintenance Domain 2175
Setting Maintenance Domain parameters 2176
Creating Maintenance Associations 2176

Configuring a CCM interval for a Maintenance Association (MA) 2177
Configuring local ports. 2177
Configuring Remote MEPs. 2178
Setting the Remote Check Start-Delay 2178
Specifying MIP creation policy 2179

Y.1731 performance management 2180
About Y.1731 2180
Y. 1731 show commands 2182

CFM monitoring and show commands 2184
Sending linktrace messages 2184
Sending loopback messages 2184
Displaying CFM configurations 2186
Displaying connectivity statistics 2188
Sample configuration for a customer's domain. 2189
Configuring CFM using Provider Bridges 2190
Displaying the connectivity status in a customer's domain 2195
Sample configuration for a customer domain using MPLS VLL 2196
Achieving end-to-end connectivity between CE1 and CE2 2197

Monitoring the status of devices in a VPLS network in a Provider's Maintenance Domain. 2206
Verifying connectivity in a VPLS network using IEEE 802.1ag 2208
Verifying connectivity in a VPLS network using IEEE 802.1ag Loopback 2211
Support for IEEE 802.1ag CFM for Provider Bridges (PB) and Provider Backbone Bridges (PBB) 2213

IEEE 802.3ah EFM-OAM 2213
EFM- OAM protocol. 2214
Process overview 2215
Link monitoring process 2216
Enabling and disabling EFM-OAM 2217
Display information 2219

Ping 2222
Executing ping 2223
Executing ping VRF 2224
Executing ping IPv6 2224

Trace route 2225
Executing traceroute 2226
Executing traceroute VRF 2226
Executing traceroute IPv6 2227

Trace-l2 protocol 2227

LSP ping and traceroute 2229
Overview 2229
LSP ping operation. 2230
LSP traceroute operation 2230

MPLS echo request 2230
MPLS echo reply 2231
LSP ping TLVs 2231
LSP FEC types 2232
Redundant RSVP LSPs 2232
One-to-one Fast ReRoute (FRR) LSPs 2232
FRR bypass LSPs 2232
Transit-originated detour 2233
LSP reoptimization 2233
PHP behavior 2233
Using the LSP ping and Traceroute commands 2233
Displaying LSP ping and traceroute statistics 2238

Chapter 61 Foundry Direct Routing and CAM Partition Profiles for the PowerConnect B-MLXe

Configuring FDR globally 2239
Configuring FDR for IPv6 routes 2239
Configuring FDR for IPv4 and IPv6 VPN routes 2239

CAM partition profiles 2240
Supernet CAM partition sharing 2241
Displaying CAM partition 2241
Displaying CAM Partition for IPv6 VPN 2244
Output from show CAM partition usage command 2244
Displaying CAM information 2247
Displaying IPv6 VPN CAM information 2248
Show cam v6acl 2248
Show IFL CAM ISID partition 2249

Configuring CAM partition size 2249

CAM overflow logging 2250
Configuring minimum logging interval and threshold value 2250

Appendix A Using Syslog

Displaying Syslog messages 2254

Configuring the Syslog service 2255
Displaying the Syslog configuration 2255
Ascending or descending option for show log command . . 2259
Disabling or re-enabling Syslog 2259
Specifying a Syslog server 2259
Specifying an additional Syslog server 2260
Disabling logging of a message level 2260
Changing the number of entries for the local buffer 2260
Changing the log facility 2261
Displaying the interface name in Syslog messages 2262
Clearing the Syslog messages from the local buffer 2262
Displaying TCP or UDP port numbers in Syslog messages . 2262
Logging all CLI commands to Syslog 2262

Syslog messages 2263

Appendix B **Global and Address Family Configuration Levels**

Accessing the address family configuration level 2306

Backward compatibility for existing BGP4 and IPv4 IS-IS configurations 2307

Global BGP4 commands and BGP4 unicast route commands 2308

Appendix C **Commands That Require a Reload**

Appendix D **Software Specifications**

IPv6 Phase II 2313

IEEE COMPLIANCE 2313

RFC COMPLIANCE 2314
- RFC compliance - BGPv4 2314
- RFC compliance - OSPF 2314
- RFC compliance - IS-IS 2314
- RFC compliance - RIP 2315
- RFC compliance - IP multicast 2315
- RFC compliance - general protocols 2315
- RFC compliance - management 2316
- RFC compliance - IPv6 management 2318
- RFC compliance - IPv6 core 2318
- RFC compliance - IPv6 routing 2318
- RFC compliance - IPv6 multicast 2319
- RFC compliance - IPv6 transitioning 2319
- RFC compliance - MPLS 2319
- RFC compliance - L3VPN 2319

Internet drafts 2320

Appendix E **Acknowledgements**

OpenSSL license 2321

Cryptographic software 2322

About This Document

Audience

This document is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Dell device, you should be familiar with the following protocols if applicable to your network – IP, RIP, OSPF, BGP, ISIS, IGMP, PIM, MPLS, and VRRP.

Supported features

Table 1 describes all of the features supported on PowerConnect B-MLXe device. Features or options not listed in Table 1 or documented in this guide are not supported.

TABLE 1 Supported features

Category	Feature description
System Level Features	
Cisco Discovery Protocol (CDP)	Allows you to configure a device to intercept and display the contents of CDP packets. This feature is useful for learning device and interface information for Cisco devices in the network.
Foundry Discovery Protocol (FDP)	Enables the devices to advertise themselves to other devices on the network.
Denial of Service (DoS) Protection	Protection from SYN attacks Protection from Smurf attacks
CLI Logging	
Management Options	Serial, Telnet and SSH access to industry-standard Command Line Interface (CLI) SSHv2 TFTP and SCP SNMP versions 1, 2, and 3
Management Options	Web GUI
High Availability	Hitless Software Upgrade Hitless Layer 2 Failover Hitless Layer 3 Failover (BGP and OSPF) IPv4 Multicast (IGMP, PIM-DM, PIM-SM, PIM-SSM)
IP Security	Dynamic ARP Inspection (DAI) DHCP Snooping DHCP with Option 82 Insertion
	IP Source Guard

TABLE 1 Supported features

Category	Feature description
Security	AAA Login Authentication using RADIUS, TACACS, TACACS+, local account, enable and line passwords AAA Enable Authentication using RADIUS, TACACS, TACACS+, local account, enable and line passwords AAA Command Authorization using RADIUS, TACACS+ AAA Command Accounting using RADIUS, TACACS+ AAA EXEC Accounting using RADIUS, TACACS+ Local passwords Secure Shell (SSH) version 2 Secure Copy (SCP) User accounts AES for SNMPv3 AES for SSHv2 Note: Telnet, SSH and SNMP servers are disabled by default, and can be enabled selectively.
Logging	Multiple SysLogD server logging
sFlow	sFlow version 5 ACL-based sFlow
Jumbo Packets	Jumbo Packet Support
Uni-Directional Link Detection (UDLD)	Monitors a link between two devices and brings the ports on both ends of the link down if the link goes down at any point between the two devices.
UDLD on tagged ports	Allows ports to send out UDLD control packets that are tagged with a specific VLAN ID as tagged UDLD control packets.
Enhanced User Password Combination	
ACL Accounting	Global statistics for inbound and outbound packets denied by ACLs
Layer 2 Features	
IEEE 802.1d	Spanning Tree Protocol (STP) Single Spanning Tree Protocol (SSTP)
IEEE 802.1p	Class of service for traffic prioritization
IEEE 802.1q	Refer to VLANs, below
IEEE 802.1w	Rapid Spanning Tree Protocol (RSTP) Single Spanning Tree Protocol (SSTP)
IEEE 802.1s	Multiple Spanning Tree Protocol
IEEE 802.1x	Port Security
IEEE 802.3ad	Link Aggregation Control Protocol
L2 ACL	Filtering based on MAC layer-2 parameters.
CPU Protection	Enhances the efficiency of the CPU on an Interface module and protects it from an excessive amount of network traffic.
MAC Port Security	
Multicast	IGMP v1, v2, v3 snooping PIM-SM snooping (IPv4 only)
	Concurrent multicast routing and snooping for IPv4

TABLE 1 Supported features

Category	Feature description
Foundry MRP	Foundry Metro Ring Protocol (MRP) Phase 1 and Phase 2
PVST or PVST+	Per-VLAN Spanning Tree (PVST)
SuperSpan	An STP enhancement that allows Service Providers (SPs) to use STP in both SP networks and customer networks.
Topology Groups	A named set of VLANs that share a Layer 2 topology. You can use topology groups with the following Layer 2 protocols: • STP • MRP • VSRP • IEEE 802.1W
Link Aggregate Groups (LAGs)	Allows you to manually configure multiple high-speed load-sharing links between two switches or routers.
Uplink-Switch	Isolated Private VLANs
VLANs	IEEE 802.1q tagging Port-based VLANs Dual-Mode VLAN Ports VLAN Transparent Hardware Flooding
VLANs	Super Aggregated VLANs (SAV) VLAN Translation Protocol-Based VLANs
VSRP	Virtual Switch Redundancy Protocol (VSRP) VSRP-fast start
VSRP - MRP Signaling	
Advanced Layer 2 Features	
IEEE 802.1ag	Connectivity Fault Management (CFM) for C-VLANs only Yes
Layer 3 Features	
IPv4 ACLs	Standard and Extended Inbound and Outbound ACL logging

TABLE 1 Supported features

Category	Feature description
BGP	BGP routes BGP peers BGP dampening BGP Confederations BGP Route Reflectors Multi-hop E-BGP Community filters Restart helper mode Multipath load sharing MD5 authentication BGP4 MIB and notifications as per RFC 4273 Multi-protocol BGP Extended Communities Route Refresh Co-operative BGP Route Filtering Graceful Restart Helper Change BGP default of BGP MED for route cost to IGP cost
	Graceful Restart
FDR	Foundry Direct Routing
IP Forwarding	IPv4 Routing IPv6 Routing Secondary Addresses
IP Static Entries	Routes ARP
IS-IS	Routes Adjacencies LSPs MD5 Authentication 3-Way Handshake for Pt-to-Pt Adjacencies BFD for IS-IS IS-IS Black Hole Avoidance PSPF Optimizations
	Traffic Engineering Extensions
IPv4 Multicast Routing	IGMP v1, v2, v3 PIM-DM PIM-SM PIM-SSM MSDP Anycast RP
Multicast	Prune Wait Timer for PIM DM

TABLE 1 Supported features

Category	Feature description
OSPF	OSPF routes OSPF adjacencies - Dynamic OFPF LSAs OSPF filtering of advertised routes MD5 authentication Restart helper mode BFD for OSPF OSPF Administrative Distance Control Using Route Maps OSPF Dynamic Metric Calculation for Trunks/VE Interfaces OSPF VRF-Lite for CE routers
	Graceful Restart Traffic Engineering (TE) Extensions
	Graceful Restart Helper
Policy-Based Routing (PBR)	
Multi-VRF	Multi-VRF for IPv4 Unicast (OSPF and Static)
	Multi-VRF for IPv4 Unicast (BGP)
RIP Versions 1 and 2	RIP routes
VRRP and VRRPE	Virtual Router Redundancy Protocol (VRRP) and VRRP Extended (VRRPE)
QoS Features	
Traffic Policing	The following rate limiting types are available on inbound and outbound ports: • Port-based • Port-and-ACL-based (Both L2 and L3 ACLs) • Hardware-based rate limiting of CPU-copied traffic
	• Port-and-priority-based • VLAN-based • VLAN-group-based (outbound only) • VLAN and priority based • VLAN-group and priority based (outbound only)
Traffic Scheduling	The following scheduling schemes are supported: • Strict Priority (SP) Scheduling
	• Weighted Fair Queuing (WFQ) scheduling • Mixed SP and WFQ scheduling
VPN Features	
Topology Groups	A named set of VLANs and VPLS endpoints that share a Layer 2 topology. You can use topology groups with the following Layer 2 protocols: • STP • MRP • VSRP • IEEE 802.1w

TABLE 1 Supported features

Category	Feature description
Layer 2 VPN	VPLS • Local VPLS (without MPLS) • VLL • Local VLL • Single Tag
	VPLS • Multicast Snooping for VPLS • IGMP and PIM Proxy for VPLS • Disabling of VPLS Local Switching • BGP Auto-Discovery • Double Tag • VPLS cpu protection
Layer 3 VPN (Requires ME_PREM plus L3_PREM licenses)	BGP/MPLS VPNs (RFC 2547bis) • OSPF Sham Link Support • Support for RFC 4382: MPLS-L3VPN Standard MIB • Multi-VRF • Per-VRID virtual MAC address assignment • Per-VRF VRRP-E • Static Routes Across VRFs
IPv6 Features	
IPv6 ACLs	Extended ACLs
IPv6 Routing Protocols	• RIPng • IS-IS for IPv6 • OSPFv3 • IPsec Authentication
	• BGP4+
IPv6 Multicast Routing	• MLD v1, v2 • PIM-SSM • PIM-SM • Anycast RP
MPLS Features	

TABLE 1 Supported features

Category	Feature description
MPLS	LDP RSVP-TE OSPF-TE ISIS-TE LSR Support OAM - LSP ping and LSR Traceroute (only on Ingress and Egress routers) VPLS and VLL Hot-standby LSPs VPLS and VLL support LAG endpoints from VPLS and VLL MPLS over LAG Fast Reroute Detour Support CSPF Adaptive LSPs
	BFD for RSVP-TE Fast Reroute Bypass Support LSP Accounting OAM - LSP ping and Traceroute

Document conventions

This section describes text formatting conventions and important notice formats used in this document.

Text formatting

The narrative-text formatting conventions that are used are as follows

bold text	Identifies command names
	Identifies the names of user-manipulated GUI elements
	Identifies keywords
	Identifies text to enter at the GUI or CLI
italic text	Provides emphasis
	Identifies variables
	Identifies document titles
`code` text	Identifies CLI output

For readability, command names in the narrative portions of this guide are presented in bold: for example, **show version**.

Command syntax conventions

Command syntax in this manual follows these conventions:

command and parameters	Commands and parameters are printed in bold.
[ ]	Optional parameter.
variable	Variables are printed in italics enclosed in angled brackets < >.
...	Repeat the previous element, for example “member[;member...]”
|	Choose from one of the parameters.

Notes, cautions, and danger notices

The following notices and statements are used in this manual. They are listed below in order of increasing severity of potential hazards.

NOTE
A note provides a tip, guidance or advice, emphasizes important information, or provides a reference to related information.

CAUTION

A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER

A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Related publications

The following Dell documents supplement the information in this guide:

- *Brocade MLXe and NetIron Family Configuration Guide*
- *PowerConnect B-MLXe Hardware Installation Guide*
- *PowerConnect B-MLXe Diagnostic Reference*

NOTE
For the latest edition of these documents, which contain the most up-to-date information, Refer to product manuals at manuals at support.dell.com.

Getting technical help or reporting errors

Dell is committed to ensuring that your investment in our products remains cost-effective. If you need assistance or find errors in the manuals, contact Dell Technical Support. When contacting Dell Technical Support have the device configuration file and an output capture of **show tech-support** command available.

Contacting Dell

For customers in the United States, call 800-WWW.DELL (800.999.3355).

NOTE
If you do not have an active Internet connection, you can find contact information on your purchase invoice, packing slip, bill, or Dell product catalog.

Dell provides several online and telephone-based support and service options. Availability varies by country and product, and some services may not be available in your area. To contact Dell for sales, technical support, or customer service issues.

1. Visit *http://www.support.dell.com*.
2. Click your country or region at the bottom of the page. For a full listing of countries and regions, click **All**.
3. In the Support menu, click **All Support**.
4. Choose the method of contacting Dell that is convenient for you.

Getting Started with the Command Line Interface 1

Overview

The following command line features are supported by NetIron MLX Series devices:

- On-Line Help
- Command Completion
- Scroll Control
- Line Editing Commands
- Accessing the CLI
- Single User in CONFIG Mode
- Multi-User Conflict During Deletion Of Group Configuration (Or Stanza)
- Searching and Filtering Output
- CLI Parsing
- Syntax Shortcuts
- Web Management Interface

This chapter presents information to help you become familiar with the PowerConnect command line interface (CLI).

As with other devices, you can manage a PowerConnect using any of the following applications:

- **Command Line Interface (CLI)** – a text-based interface accessible directly from a PC or terminal attached to the management module's serial (Console) port or 10BaseT/100BaseTX Ethernet (management) port, or from a Telnet connection to the PC or terminal.
- **Web Management Interface** – a GUI-based management interface accessible through an HTTP (web browser) connection.
- **SNMP Network Manager** – an optional SNMP-based standalone GUI application.

This user guide describes how to configure the features using the CLI. This chapter how to use the CLI.

NOTE
This user guide assumes that an IP address and default gateway have been assigned to the PowerConnect when it was installed. If you need to assign an IP address or default gateway to the device, refer to the PowerConnect Installation guides.

Logging on through the CLI

After an IP address is assigned to the device's management port, you can access the CLI through a PC or terminal attached to the management module's serial (Console) port or 10BaseT/100BaseTX Ethernet (management) port, or from a Telnet or SSH connection to the PC or terminal.

You can initiate a local Telnet, SSH or SNMP connection by specifying the management port's IP address.

The commands in the CLI are organized into the following levels:

- **User EXEC** – Lets you display information and perform basic tasks such as pings and traceroutes.
- **Privileged EXEC** – Lets you use the same commands as those at the User EXEC level plus configuration commands that do not require saving the changes to the system-config file.
- **CONFIG** – Lets you make configuration changes to the device. To save the changes across software reloads and system resets, you need to save them to the system-config file. The CONFIG level contains sub-levels for individual ports, for VLANs, for routing protocols, and other configuration areas.

NOTE
By default, the PowerConnect devices have all management access disabled, except for console port management. To create access, you must configure Enable passwords or local user accounts, or you can configure the device to use a RADIUS or TACACS or TACACS+ server for authentication.

On-line help

To display a list of available commands or command options, enter "?" or press Tab. If you have not entered part of a command at the command prompt, all the commands supported at the current CLI level are listed. If you enter part of a command, then enter "?" or press Tab, the CLI lists the options you can enter at this point in the command string.

If you enter an invalid command, a message appears indicating the command was unrecognized.

Example

```
NetIron(config)# rooter ip
Unrecognized command
```

Command completion

The CLI supports command completion, so you do not need to enter the entire name of a command or option. As long as you enter enough characters of the command or option name to avoid ambiguity with other commands or options, the CLI understands what you are typing.

Scroll control

By default, the CLI uses a page mode to paginate displays that are longer than the number of rows in your terminal emulation window. For example, if you display a list of all the commands at the global CONFIG level but your terminal emulation window does not have enough rows to display them all at once, the page mode stops the display and lists your choices for continuing the display.

Example

```
aaa
access-list
all-client
arp
banner
base-mac-addr
boot
some lines omitted for brevity...

 default-vlan-id
 enable
 enable-acl-counter
 end
 exit
--More--, next page: Space, next line: Return key, quit: Control-c
```

The software provides the following scrolling options:

- Press the Space bar to display the next page (one screen at time).
- Press the Return or Enter key to display the next line (one line at a time).
- Press Ctrl-C cancel the display.

Line editing commands

The CLI supports the following line editing commands. To enter a line-editing command, use the CTRL-key combination for the command by pressing and holding the CTRL key, then pressing the letter associated with the command.

TABLE 2 CLI line-editing commands

Ctrl-key combination	Description
Ctrl-A	Moves to the first character on the command line.
Ctrl-B	Moves the cursor back one character.
Ctrl-C	Escapes and terminates command prompts and ongoing tasks (such as lengthy displays), and displays a fresh command prompt.
Ctrl-D	Deletes the character at the cursor.
Ctrl-E	Moves to the end of the current command line.
Ctrl-F	Moves the cursor forward one character.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-L; Ctrl-R	Repeats the current command line on a new line.
Ctrl-N	Enters the next command line in the history buffer.
Ctrl-P	Enters the previous command line in the history buffer.
Ctrl-U; Ctrl-X	Deletes all characters from the cursor to the beginning of the command line.

TABLE 2 CLI line-editing commands (Continued)

Ctrl-key combination	Description
Ctrl-W	Deletes the last word you typed.
Ctrl-Z	Moves from any CONFIG level of the CLI to the Privileged EXEC level; at the Privileged EXEC level, moves to the User EXEC level.

EXEC commands

There are two different levels of EXEC commands, the ***User Level*** and the ***Privileged Level***.

User level

The User level commands are at the top of the CLI hierarchy. These are the first commands that you have access to when connected to the device through the CLI. For example, when you first connect to the PowerConnect, you may see one of the following prompts.

```
NetIron MLXe>
```

The "PowerConnect" part of the prompt is configurable. Your system may display a different string.

At this level, you can view basic system information and verify connectivity but cannot make any changes to the device configuration. To make changes to the configuration, you must move to other levels of the CLI hierarchy. such as the Privileged EXEC level.

Privileged EXEC level

Commands at the Privileged EXEC level enable you to transfer and store software images and configuration files between the network and the system, and review the configuration.

You reach this level by entering the **enable** [<*password*>] or **enable** <*username*> <*password*> at the User EXEC level.

Example

```
NetIron>enable
```

or

```
NetIron>enable user1 mypassword
```

After entering the enable command, you see the following prompt.

```
NetIron>#
```

The prompt indicates that you are at the Privilege EXEC level.

When you are at the Privilege EXEC level, you can enter commands that are available at that level. It is also at this level where you enter the **configure terminal** command to Global Configuration level.

Global level

The global CONFIG level allows you to globally apply or modify parameters for ports on the device. You reach this level by entering **configure terminal** at the privileged EXEC level.

```
NetIron> enable
NetIron># configure terminal
```

The prompt changes to the Global Configuration level.

```
NetIron(config)#
```

CONFIG commands

CONFIG commands modify the configuration of a PowerConnect. When you are at the Global Configuration level, you can enter commands to configure the features in a PowerConnect. This section describes the CONFIG CLI levels.

Redundancy level

This redundancy level allows you to configure redundancy parameters for redundant management modules. You reach this level by entering the **redundancy** command at the global CONFIG level.

Interface level

The interface level allows you to assign or modify specific port parameters on a specific port. You reach this level by entering the following at the global CONFIG level:

- **interface ethernet** <*slot/port*>
- **interface loopback** <*num*>
- **interface management** <*portnum*>
- or **interface ve** <*num*>
- **interface tunnel** <*tunnel_id*>
- **interface group-ve** <*vlan_group_id*>

LAG level

The LAG level allows you to change parameters for statically-configured LAG groups. You reach this level by entering a **LAG** command with the appropriate port parameters.

Router RIP level

The RIP level allows you to configure parameters for the RIP routing protocol. You reach this level by entering the **router rip** command at the global CONFIG level.

Router OSPF level

The OSPF level allows you to configure parameters for the OSPF routing protocol. You reach this level by entering the **router ospf** command at the global CONFIG level.

BGP level

The BGP level allows you to configure Border Gateway Protocol version 4 (BGP4) features. You reach this level by entering the **router bgp** command at the global CONFIG level.

Global BGP and BGP4 unicast address family level

The global BGP and BGP4 unicast address family levels are present only on devices that support IPv6. The global BGP level allows you to configure the BGP routing protocol. The BGP4 unicast address family level allows you to configure a BGP4 unicast route. For backward compatibility, you can currently access BGP4 unicast address family commands at both global BGP configuration and BGP4 unicast address family configuration levels. Therefore, the global BGP and BGP4 unicast address family commands are documented together.

You reach the global BGP level by entering the **router bgp** command at the global CONFIG level. You reach the BGP4 unicast address family level by entering the **address-family ipv4 unicast** command at the global BGP level.

BGP4 multicast address family level

The BGP4 multicast address family level allows you to configure BGP4 multicast routes. You reach this level by entering the **address-family ipv4 multicast** command at the global BGP, BGP4 unicast address family, or IPv6 BGP unicast address family levels.

Router DVMRP level

The DVMRP level allows you to configure details for the DVMRP multicast protocol. You reach this level by entering the **router dvmrp** command at the global CONFIG level.

Router PIM level

The PIM level allows you to configure parameters for the Protocol Independent Multicast (PIM) routing protocol. You reach this level by entering the **router pim** command at the global CONFIG level.

Route Map level

The Route Map level allows you to configure parameters for a BGP4 route map. You reach this level by entering the **route-map** <*name*> command at the global CONFIG level.

Router VRRP level

The VRRP level allows you to configure parameters for the Virtual Router Redundancy Protocol (VRRP). You reach this level by entering the **router vrrp** command at the global CONFIG level, then entering the **ip vrrp vrid** <*num*> command at the interface configuration level.

Router VRRPE level

The VRRPE level allows you to configure parameters for VRRP Extended. You reach this level by entering the **router vrrp-extended** command at the global CONFIG level, then entering the **ip vrrp-extended vrid** <*num*> command at the interface configuration level.

VLAN level

Policy-based VLANs allow you to assign VLANs to a protocol, port, or 802.1q tags.

You reach this level by entering the **vlan** <*vlan-id*> command at the Global CONFIG Level.

Ethernet Service Instance (ESI) level

Ethernet Service Instance (ESI) allow you to assign an ESI to a protocol, or port.

Metro ring level

Metro rings provide Layer 2 connectivity and fast failover in ring topologies.

You reach this level by entering the metro-ring <*ring-id*> command at the VLAN CONFIG Level.

VSRP level

The VSRP level allows you to configure parameters for the Virtual Switch Redundancy Protocol (VSRP). You reach this level by entering the **vsrp vrid** <*num*> command at the VLAN configuration level, then entering the **vsrp vrid** <*num*> command at the VLAN configuration level.

Topology group level

A topology group enables you to control the Layer 2 protocol configuration and Layer 2 state of a set of ports in multiple VLANs based on the configuration and states of those ports in a single master VLAN. One instance of the Layer 2 protocol controls all the VLANs.

You reach this level by entering the **topology-group** <*group-id*> command at the Global CONFIG Level.

802.1X port security level

The 802.1X port security level allows you to configure the 802.1X port security. You reach this level by entering the **dot1x-enable** command at the Global level.

MAC port security level

The MAC port security level allows you to configure the port security feature. You reach this level by entering the **port security** command at the at the Global or Interface levels.

Accessing the CLI

The CLI can be accessed through both serial and Telnet connections. For initial log on, you must use a serial connection. Once an IP address is assigned, you can access the CLI through Telnet.

Once connectivity to the device is established, you will see the a prompt.

```
NetIron>
```

When accessing the CLI through Telnet, you maybe prompted for a password. By default, the password required is the password you enter for general access at initial setup. You also have the option of assigning a separate password for Telnet access with the **enable telnet password** <*password*> command, found at the Global Level.

At initial log on, all you need to do is type **enable** at the prompt, then press Return. You only need to enter a password after a permanent password is entered at the Global CONFIG Level of the CLI.

To reach the Global CONFIG Level, the uppermost level of the CONFIG commands, enter the following commands

NetIron> enable	User Level commands
NetIron# configure terminal	Privileged Level-EXEC commands
NetIron(config)#	Global Level-CONFIG commands

You can then reach all other levels of the CONFIG command structure from this point.

The CLI prompt will change at each level of the CONFIG command structure, to easily identify the current level.

```
NetIron>    User Level EXEC Command
NetIron#    Privileged Level EXEC Command
NetIron(config)#Global Level CONFIG Command
NetIron(config-if-e10000-5/1)#Interface Level CONFIG Command
NetIron(config-lbif-1)#Loopback Interface CONFIG Command
NetIron(config-ve-1)#Virtual Interface CONFIG Command
NetIron(config-trunk-4/1-4/8)# trunk group CONFIG Command
NetIron(config-if-e10000-tunnel)#IP Tunnel Level CONFIG Command
NetIron(config-bgp-router)#BGP Level CONFIG Command
NetIron(config-dvmrp-router)#DVMRP Level CONFIG Command
NetIron(config-ospf-router)#OSPF Level CONFIG Command
NetIron(config-isis-router)#IS-IS Level CONFIG Command
NetIron(config-pim-router)#PIM Level CONFIG Command
NetIron(config-redundancy)#Redundant Management Module CONFIG Command
NetIron(config-rip-router)#RIP Level CONFIG Command
NetIron(config-port-80)#Application Port CONFIG Command
NetIron(config-bgp-routemap Map_Name)#Route Map Level CONFIG Command
NetIron(config-vlan-1)#VLAN Port-based Level CONFIG Command
NetIron(config-vlan-atalk-proto)#VLAN Protocol Level CONFIG Command
```

NOTE

The CLI prompt at the interface level includes the port speed. The speed is one of the following:
NetIron(config-if–e100-5/1)# – The interface is a 10/100 port.
NetIron(config-if–e1000-5/1)# – The interface is a Gigabit port.
For simplicity, the port speeds sometimes are not shown in example Interface level prompts in this manual.

Single user in CONFIG mode

By default, more than one user can enter the CONFIG mode of a device CLI, which is accessed through the **configure terminal** command. While in CONFIG mode, users can override another user's configuration changes.

You can configure a device to allow only one user to be in CONFIG mode at any one time. Other users who try to enter that mode in will be denied. To allow only one user to enter CONFIG mode, enter the following command.

```
NetIron#configure terminal
NetIron(config)# single-config-user
NetIron(config)# write memory
```

Syntax: [no] single-config-user

After the **single-config-user** command is issued, the device will not allow more than one user to enter CONFIG mode. However, if you run the command while more than one user is in CONFIG mode, the other users continue to be in CONFIG mode and can potentially override each other's configuration changes. Only users who try to enter the CONFIG mode after the command is issued are prevented from entering CONFIG mode. If a user is already in that mode and another user tries to enter CONFIG mode after the **single-config-user** command is issued, the following error is displayed.

```
NetIron#configure terminal
Single user config mode is being enforced. Config mode is being used by
<session-type> session.
```

where <*session-type*> can be one of the following:

- console
- telnet <*number*>
- SSH <*number*>

Multi-user conflict during deletion of group configuration (or stanza)

By default, a user may delete a group configuration, even if another user is simultaneously in that mode. You can disable this feature by issuing the **enable multi-user-mode-deletion** command.

To allow only one user to delete group configurations, enter the following command.

```
NetIron#configure terminal
NetIron(config)# enable multi-user-mode-deletion
NetIron(config)# write memory
```

When a user attempts to delete a group configuration from the CLI, and another user is already within that group configuration, the user who tries to delete a group configuration in that mode will be denied and will receive the following error message.

```
Session 1:
NetIron(config)# vlan 10
NetIron(config-vlan-10)#
Session 2:
NetIron(config)# no vlan 10
"Error: Cannot undo the configuration as {console|telnet|SSH} session is
using this mode."
```

Syntax: [no] enable multi-user-mode-deletion

Use the **no** form of this command will allow multiple users the ability to delete group configurations.

NOTE
This feature will not work on commands that are issued from the WEB management and the SNMP management.

Navigating among command levels

To reach other CLI command levels, you need to enter certain commands. At each level there is a launch command that allows you to move either up or down to the next level.

CLI command structure

Many CLI commands may require textual or numeral input as part of the command.

Required or optional fields

These fields are either required or optional depending on how the information is bracketed. For clarity, a few CLI command examples are explained below.

Example

Syntax: **[no] deny redistribute** <*value*> **all | bgp | rip | static address** <*ip-addr*> <*ip-mask*> **[match-metric** <*value*> **| set-metric** <*value*>**]**

When an item is bracketed with *"< >"* symbols, the information requested is a variable and required.

When an item is not enclosed by *"< >"* or "[ ]" symbols, the item is a required keyword.

When an item is bracketed with "[ ]" symbols, the information requested is optional.

Optional fields

When two or more options are separated by a vertical bar, " | ", you must enter one of the options as part of the command.

Example

Syntax: **priority normal | high**

For example, the "normal | high" entry in the Syntax above means that priority can be either priority normal or priority high. The command in the syntax above requires that you enter either normal or high as part of the command.

List of available options

To get a quick display of available options at a CLI level or for the next option in a command string, enter a question mark (?) at the prompt or press TAB.

Example

To view all available commands at the user EXEC level, enter the following or press TAB at the User EXEC CLI level.

```
NetIron> ? <return>
enable
exit
fastboot
ping
show
stop-trace-route
traceroute
```

You also can use the question mark (?) with an individual command, to see all available options or to check context.

Example

Enter the following to view possible **copy** command options.

```
NetIron# copy ?
  flash
  running-config
  startup-config
  tftp
NetIron# copy flash ?
  tftp
```

Searching and filtering output

You can filter CLI output from **show** commands and at the --More-- prompt. You can search for individual characters, strings, or construct complex regular expressions to filter the output.

Searching and filtering output from show commands

You can filter output from **show** commands to display lines containing a specified string, lines that do not contain a specified string, or output starting with a line containing a specified string. The search string is a regular expression consisting of a single character or string of characters. You can use special characters to construct complex regular expressions. refer to "Using special characters in regular expressions" on page 13 for information on special characters used with regular expressions.

Displaying lines containing a specified string

The following command filters the output of the **show interface** command for port 3/11 so it displays only lines containing the word "Internet". This command can be used to display the IP address of the interface.

```
NetIron# show interface e 3/11 | include Internet
  Internet address is 192.168.1.11/24, MTU 1518 bytes, encapsulation ethernet
```

Syntax: <*show-command*> **| include** <*regular-expression*>

NOTE
The vertical bar (|) is part of the command.

Note that the regular expression specified as the search string is case sensitive. In the example above, a search string of "Internet" would match the line containing the IP address, but a search string of "internet" would not.

Displaying lines that do not contain a specified string

The following command filters the output of the **show who** command so it displays only lines that do not contain the word "closed". This command can be used to display open connections to the device.

```
NetIron# show who | exclude closed
Console connections:
        established
        you are connecting to this session
        2 seconds in idle
Telnet connections (inbound):
```

```
 1      established, client ip address 192.168.9.37
        27 seconds in idle
Telnet connection (outbound):
SSH connections:
```

Syntax: <*show-command*> **| exclude** <*regular-expression*>

Displaying lines starting with a specified string

The following command filters the output of the **show who** command so it displays output starting with the first line that contains the word "SSH". This command can be used to display information about SSH connections to the PowerConnect.

```
NetIron# show who | begin SSH
SSH connections:
 1      established, client ip address 192.168.9.210
        7 seconds in idle
 2      closed
 3      closed
 4      closed
 5      closed
```

Syntax: <*show-command*> **| begin** <*regular-expression*>

Searching and filtering output at the --More-- prompt

The --More-- prompt is displayed when output extends beyond a single page. From this prompt, you can press the Space bar to display the next page, the Return or Enter key to display the next line, or Ctrl-C or Q to cancel the display. You can also search and filter output from this prompt.

Example

```
NetIron# ?
  append               Append one file to another
  attrib               Change file attribute
  boot                 Boot system from bootp/tftp server/flash image
  cd                   Change current working directory
  chdir                Change current working directory
  clear                Clear table/statistics/keys
  clock                Set clock
  configure            Enter configuration mode
  copy                 Copy between flash, tftp, config/code
  cp                   Copy file commands
  debug                Enable debugging functions (see also 'undebug')
  delete               Delete file on flash
  dir                  List files
  dm                   test commands
  dot1x                802.1X
  erase                Erase image/configuration files from flash
  exit                 Exit Privileged mode
  fastboot             Select fast-reload option
  force-sync-standby   Sync active flash (pri/sec/mon/startup config/lp images)
                       to standby
  format               Format PCMCIA card
  hd                   Hex dump
  ipc                  IPC commands
--More--, next page: Space, next line: Return key, quit: Control-c
```

At the --More-- prompt, you can press the forward slash key (/) and then enter a search string. The device displays output starting from the first line that contains the search string, similar to the **begin** option for **show** commands.

Example

```
--More--, next page: Space, next line: Return key, quit: Control-c
/telnet
```

The results of the search are displayed.

```
searching...
telnet               Telnet by name or IP address
terminal             Change terminal settings
traceroute           TraceRoute to IP node
undelete             Recover deleted file
whois                WHOIS lookup
write                Write running configuration to flash or terminal
```

To display lines containing only a specified search string (similar to the **include** option for **show** commands) press the plus sign key (+) at the --More-- prompt and then enter the search string.

```
--More--, next page: Space, next line: Return key, quit: Control-c
+telnet
```

The filtered results are displayed.

```
filtering...
  telnet              Telnet by name or IP address
```

To display lines that do not contain a specified search string (similar to the **exclude** option for **show** commands) press the minus sign key (-) at the --More-- prompt and then enter the search string.

```
--More--, next page: Space, next line: Return key, quit: Control-c
-telnet
```

The filtered results are displayed.

```
filtering...
sync-standby         Sync active flash (pri/sec/mon/startup config/lp images)
                     to standby if different
terminal             Change terminal settings
traceroute           TraceRoute to IP node
undelete             Recover deleted file
whois                WHOIS lookup
write                  Write running configuration to flash or terminal
```

As with the commands for filtering output from **show** commands, the search string is a regular expression consisting of a single character or string of characters. You can use special characters to construct complex regular expressions. Refer to the next section for information on special characters used with regular expressions.

Using special characters in regular expressions

You use a regular expression to specify a single character or multiple characters as a search string. In addition, you can include special characters that influence the way the software matches the output against the search string. These special characters are listed in the following table.

TABLE 3 Special characters for regular expressions

Character	Operation
.	The period matches on any single character, including a blank space. For example, the following regular expression matches “aaz”, “abz”, “acz”, and so on, but not just “az”: a.z
*	The asterisk matches on zero or more sequential instances of a pattern. For example, the following regular expression matches output that contains the string “abc”, followed by zero or more Xs: abcX*
+	The plus sign matches on one or more sequential instances of a pattern. For example, the following regular expression matches output that contains "de", followed by a sequence of “g”s, such as “deg”, “degg”, “deggg”, and so on: deg+
?	The question mark matches on zero occurrences or one occurrence of a pattern. For example, the following regular expression matches output that contains "dg" or "deg": de?g **NOTE:** Normally when you type a question mark, the CLI lists the commands or options at that CLI level that begin with the character or string you entered. However, if you enter Ctrl-V and then type a question mark, the question mark is inserted into the command line, allowing you to use it as part of a regular expression.
^	A caret (when not used within brackets) matches on the beginning of an input string. For example, the following regular expression matches output that begins with “deg”: ^deg
$	A dollar sign matches on the end of an input string. For example, the following regular expression matches output that ends with “deg”: deg$
_	An underscore matches on one or more of the following: • , (comma) • { (left curly brace) • } (right curly brace) • ((left parenthesis) •) (right parenthesis) • The beginning of the input string • The end of the input string • A blank space For example, the following regular expression matches on “100” but not on “1002”, “2100”, and so on. _100_
[ ]	Square brackets enclose a range of single-character patterns. For example, the following regular expression matches output that contains “1”, “2”, “3”, “4”, or “5”: [1-5] You can use the following expression symbols within the brackets. These symbols are allowed only inside the brackets. • ^ – The caret matches on any characters except the ones in the brackets. For example, the following regular expression matches output that does not contain “1”, “2”, “3”, “4”, or “5”: [^1-5] • - The hyphen separates the beginning and ending of a range of characters. A match occurs if any of the characters within the range is present. Refer to the example above.

TABLE 3 Special characters for regular expressions (Continued)

Character	Operation
|	A vertical bar separates two alternative values or sets of values. The output can match one or the other value. For example, the following regular expression matches output that contains either “abc” or “defg”: abc|defg
( )	Parentheses allow you to create complex expressions. For example, the following complex expression matches on “abc”, “abcabc”, or “defg”, but not on “abcdefgdefg”: ((abc)+)|((defg)?)

If you want to filter for a special character instead of using the special character as described in the table above, enter “\” (backslash) in front of the character. For example, to filter on output containing an asterisk, enter the asterisk portion of the regular expression as “*”.

```
NetIron# show ip route bgp | include \*
```

Allowable characters for LAG names

When creating a LAG name, you can use spaces in a file or subdirectory name if you enclose the name in double quotes. For example, to specify a subdirectory name that contains spaces, enter a string such as the following: “a long subdirectory name”. The maximum length for a string is 64 characters.

The following characters are valid in file names:

- All upper and lowercase letters
- All digits

Any of the following special characters are valid:

- $
- %
- '
- -
- _
- @
- ~
- `
- !
- (
-)
- {
- }
- ^
- #
- &

CLI parsing enhancement

The response to an invalid keyword, the command returns to the cursor will include all valid content up to where the error was made. The prompt will only delete the invalid keyword "proc" and return to a prompt with the command "PowerConnect# show". This will allow the user to continue typing from the point of failure, rather than having to type out the entire command again.

```
NetIron# show proc
Unrecognized command
NetIron# show
```

Syntax shortcuts

A command or parameter can be abbreviated as long as enough text is entered to distinguish it from other commands at that level. For example, given the possible commands **copy tftp**... and **config tftp**..., possible shortcuts are **cop tftp** and **con tftp** respectively. In this case, **co** does not properly distinguish the two commands.

Saving configuration changes

You can make configuration changes while the device is running. The type of configuration change determines whether or not it becomes effective immediately or requires a save to flash (**write memory**) and reset of the system (**reload**), before it becomes active.

This approach in adopting configuration changes:

- Allows you to make configuration changes to the operating or running configuration of the device to address a short-term requirement or validate a configuration without overwriting the permanent configuration file, the startup configuration, that is saved in the system flash, and;
- Ensures that dependent or related configuration changes are all cut in at the same time.

In all cases, if you want to make the changes permanent, you need to save the changes to flash using the **write memory** command. When you save the configuration changes to flash, this will become the configuration that is initiated and run at system boot.

NOTE
Most configuration changes are dynamic and thus do not require a software reload. If a command requires a software reload to take effect, the documentation states this.

Modifying startup and running configuration file manually

When you manually modify a **startup-config** or **running-config** file, ensure that you do not delete the (**!**) from any of the lines in the configuration file.

Securing Access to Management Functions 2

Overview

The following Security features are supported by NetIron MLX Series devices:

- Restricting Remote Access to Management Functions
- Setting Up Local User Accounts
- Web Management Interface
- SSL Security for the Web Management Interface
- TACACS or TACACS+ Security
- New encryption code for passwords, authentication keys, and community strings
- RADIUS Security
- Interactive multi-factor RADIUS security support (e.g., for RSA SecurID)
- AAA on the Console
- AAA Authentication-Method Lists
- AES Encryption for SNMPv3
- AES Encryption for SSHv2

By default, the PowerConnect devices have all management access disabled. This chapter explains how to secure access to management functions on the PowerConnect devices. It contains the following sections:

- "Securing access methods" on page 18 lists the management access methods available on the PowerConnect devices and the ways you can secure each one
- "Restricting remote access to management functions" on page 20 explains how to restrict access to management functions from remote sources, including Telnet, the Web Management Interface, and SNMP
- "Setting passwords" on page 26 explains how to set passwords for Telnet access and management privilege levels
- "Setting up local user accounts" on page 31 explains how to define user accounts to regulate who can access management functions.
- "Configuring TACACS or TACACS+ security" on page 39 explains how to configure TACACS or TACACS+ authentication, authorization, and accounting.
- "Configuring RADIUS security" on page 55 explains how to configure RADIUS authentication, authorization, and accounting.
- "Configuring AAA on the console" on page 72
- "Configuring authentication-method lists" on page 73 explains how to set the order that authentication methods are consulted when more than one is used with an access method.

NOTE
For the PowerConnect devices, RADIUS Challenge is supported for 802.1x authentication but not for login authentication. Also, multiple challenges are supported for TACACS+ login authentication.

Securing access methods

The following table lists the management access methods available on the PowerConnect devices, how they are secured by default, and the ways in which they can be secured.

TABLE 4 Ways to secure management access to the PowerConnect devices

Access method	How the access method is secured by default	Ways to secure the access method	Refer to page
Serial access to the CLI	Not secured	Establish passwords for management privilege levels Establish username and password to log in to the console.	page 27 page 72
Access to the Privileged EXEC and CONFIG levels of the CLI	Not secured	Establish a password for Telnet access to the CLI	page 27
		Establish passwords for management privilege levels	page 27
		Set up local user accounts	page 31
		Configure TACACS or TACACS+ security	page 39
		Configure RADIUS security	page 55
Telnet access Telnet server is turned off by default.		Regulate Telnet access using ACLs	page 20
		Allow Telnet access only from specific IP addresses	page 23
		Allow Telnet access only to clients connected to a specific VLAN	page 25
		Disable Telnet access	page 26
		Establish a password for Telnet access	page 27
		Establish passwords for privilege levels of the CLI	page 27
		Set up local user accounts	page 31
		Configure TACACS or TACACS+ security	page 39
		Configure RADIUS security	page 55

TABLE 4 Ways to secure management access to the PowerConnect devices (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	Refer to page
Secure Shell (SSH) access For more information on SSH, refer to Chapter 48, "Configuring Secure Shell and Secure Copy"	Not configured	Configure SSH	page 2015
		Disable SSH server.	
		Set up access using DSA Challenge-Response Authentication (excludes use of username and password credentials)	page 2018
		Regulate SSH access using ACLs	page 21
		Allow SSH access only from specific IP addresses	page 23
		Establish passwords for privilege levels of the CLI	page 27
		Set up local user accounts	page 31
		Configure TACACS or TACACS+ security	page 39
		Configure RADIUS security	page 55
Web management access	SNMP read or read-write community strings Web server is turned off by default.		
		Configure SSL security for the Web Management Interface	page 37
		Set up local user accounts	page 31
		Establish SNMP read or read-write community strings for SNMP versions 1 and 2	page 23
		Configure AAA command for Web access	
		Configure TACACS or TACACS+ security	page 39
		Configure RADIUS security	page 55
SNMP access	SNMP read or read-write community strings and the password to the Super User privilege level **NOTE:** SNMP read or read-write community strings are always required for SNMP access to the device. SNMP access is disabled by default.	Regulate SNMP access using ACLs	page 22
		Allow SNMP access only from specific IP addresses	page 23
		Disable SNMP access	page 26
		Allow SNMP access only to clients connected to a specific VLAN	page 25
		Establish passwords to management levels of the CLI	page 27
		Set up local user accounts	page 31
		Configure AAA command for SNMP access	
		Establish SNMP read or read-write community strings	page 39
TFTP access	Not secured	Allow TFTP access only to clients connected to a specific VLAN	page 25

Restricting remote access to management functions

You can restrict access to management functions from remote sources, including Telnet, and SNMP. The following methods for restricting remote access are supported:

- Using ACLs to restrict Telnet, SSH or SNMP access
- Allowing remote access only from specific IP addresses
- Allowing remote access only to clients connected to a specific VLAN
- Specifically disabling Telnet, SSH, Web Management Interface, or SNMP access to the device

Using ACLs to restrict remote access

You can use ACLs to control the following access methods to management functions on the PowerConnect router:

- Telnet access
- SSH access
- SNMP access

NOTE
IP ACLs are IP version specific. When both IPv4 and IPv6 ACLs are configured, the IPv4 ACL will be applied to sessions from IPv4 clients and the IPv6 ACL will be applied to sessions from IPv6 clients.

Follow the steps listed below to configure access control for these management access methods.

1. Configure an ACL with the IP addresses you want to allow to access the device. You can specify an IPv6 ACL, a numbered standard IPv4 ACL, or a named standard IPv4 ACL.
2. Configure a Telnet access group, SSH access group, web access group, and SNMP community strings for SNMPv1, SNMPv2c or SNMPv3 user. Each of these configuration items accepts an ACL as a parameter. The ACL contains entries that identify the IP addresses that can use the access method.

The following sections present examples of how to secure management access using ACLs. Refer to "Access Control List" on page 785 and "Configuring an IPv6 Access Control List" on page 1769 for more information on configuring ACLs.

NOTE
ACL filtering for remote management access is done in hardware.

Using an ACL to restrict Telnet access

To configure an ACL that restricts Telnet access to the device, enter commands such as the following:

```
NetIron(config)# access-list 10 deny host 209.157.22.32
NetIron(config)# access-list 10 deny 209.157.23.0 0.0.0.255
NetIron(config)# access-list 10 deny 209.157.24.0 0.0.0.255
NetIron(config)# access-list 10 deny 209.157.25.0/24
NetIron(config)# access-list 10 permit any
NetIron(config)# telnet access-group 10
NetIron(config)# write memory
```

The commands configure ACL 10, then apply it as the access list for Telnet access. The device allows Telnet access to all IP addresses except those listed in ACL 10.

Syntax: [no] telnet access-group {<*num*> | <*name*> **| ipv6** <*ipv6-acl-name*>}

Use the **ipv6** parameter if you are applying an IPv6 access list.

The <*num*> variable specifies the number of a standard IPv4 ACL, 1 – 99.

The <*name*> variable specifies the standard IPv4 access list name.

The <*ipv6-acl-name*> variable specifies the IPv6 access list name.

NOTE
ACLs for Telnet sessions will be applied only to inbound sessions.

To configure a more restrictive ACL, create permit entries and omit the **permit any** entry at the end of the ACL.

Example

```
NetIron(config)# access-list 10 permit host 209.157.22.32
NetIron(config)# access-list 10 permit 209.157.23.0 0.0.0.255
NetIron(config)# access-list 10 permit 209.157.24.0 0.0.0.255
NetIron(config)# access-list 10 permit 209.157.25.0/24
NetIron(config)# telnet access-group 10
NetIron(config)# write memory
```

The ACL in the example permits Telnet access only from the IPv4 addresses in the **permit** entries and denies Telnet access from all other IP addresses.

Using an ACL to restrict SSH access

To configure an ACL that restricts SSH access to the device, enter commands such as the following:

```
NetIron(config)# access-list 12 deny host 209.157.22.98
NetIron(config)# access-list 12 deny 209.157.23.0 0.0.0.255
NetIron(config)# access-list 12 deny 209.157.24.0/24
NetIron(config)# access-list 12 permit any
NetIron(config)# ssh access-group 12
NetIron(config)# write memory
```

Syntax: [no] ssh access-group {<*num*> **|** <*name*> **| ipv6** <*ipv6-acl-name*>}

Use the **ipv6** parameter if you are applying an IPv6 access list.

The <*num*> variable specifies the number of a standard IPv4 ACL, 1 – 99.

The <*name*> variable specifies the standard IPv4 access list name.

The <*ipv6-acl-name*> variable specifies the IPv6 access list name.

These commands configure ACL 12, then apply the ACL as the access list for SSH access. The device denies SSH access from the IPv4 addresses listed in ACL 12 and permits SSH access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny SSH access from all IP addresses.

NOTE
In this example, the **command ssh access-group 10** could have been used to apply the ACL configured in the example for Telnet access. You can use the same ACL multiple times.

Using ACLs to restrict SNMP access

To restrict SNMP access to the device using ACLs, enter commands such as the following.

NOTE
The syntax for using ACLs for SNMP access is different from the syntax for controlling Telnet, SSH, and Web management access using ACLs.

```
NetIron(config)# access-list 25 deny host 209.157.22.98
NetIron(config)# access-list 25 deny 209.157.23.0 0.0.0.255
NetIron(config)# access-list 25 deny 209.157.24.0 0.0.0.255
NetIron(config)# access-list 25 permit any
NetIron(config)# access-list 30 deny 209.157.25.0 0.0.0.255
NetIron(config)# access-list 30 deny 209.157.26.0/24
NetIron(config)# access-list 30 permit any
NetIron(config)# snmp-server community public ro 25
NetIron(config)# snmp-server community private rw 30
NetIron(config)# write memory
```

These commands configure ACLs 25 and 30, then apply the ACLs to community strings. ACL 25 is used to control read-only access using the "public" community string. ACL 30 is used to control read-write access using the "private" community string.

Syntax: [no] **snmp-server community** <*string*> **{ro | rw} {**<*standard-acl-name*> **|** <*standard-acl-id*> **| ipv6** <*ipv6-acl-name*>**}**

The <*string*> variable specifies the SNMP community string the user must enter to gain SNMP access.

The **ro** parameter indicates that the community string is for read-only ("get") access. The **rw** parameter indicates the community string is for read-write ("set") access.

The **ipv6** parameter indicates that you are applying an IPv6 access list.

The <*standard-acl-name*> or <*standard-acl-id*> or <*ipv6-acl-name*> variable specifies which ACL will be used to filter incoming SNMP packets.

The <*standard-acl-id*> variable specifies the number of a standard IPv4 ACL, 1 – 99.

The <*standard-acl-name*> variable specifies the standard IPv4 access list name.

The <*ipv6-acl-name*> variable specifies the IPv6 access list name.

NOTE
When **snmp-server community** is configured, all incoming SNMP packets are validated first by their community strings and then by their bound ACLs. Packets are permitted if no filters are configured for an ACL.

Defining the console idle time

By default, a PowerConnect router does not time out serial console sessions. A serial session remains open indefinitely until you close it. You can however define how many minutes a serial management session can remain idle before it is timed out.

To configure the idle time for a serial console session, use the following command.

```
NetIron(config)# console timeout 120
```

Syntax: [no] **console timeout** <*0 – 240*>

- Possible values: 0 – 240 minutes
- Default value: 0 minutes (no timeout)

NOTE
The standard for the idle-timeout RADIUS attribute is for it to be implemented in seconds as opposed to the minutes that the PowerConnect router uses. If this attribute is used for setting idle time instead of this configuration, the value from the idle-timeout RADIUS attribute will be converted from seconds to minutes and truncated to the nearest minute.

Restricting remote access to the device to specific IP addresses

By default, a PowerConnect router does not control remote management access based on the IP address of the managing device. You can restrict remote management access to a single IP address for the following access methods:

- Telnet access
- SNMP access

In addition, if you want to restrict all three access methods to the same IP address, you can do so using a single command.

The following examples show the CLI commands for restricting remote access. You can specify only one IP address with each command. However, you can enter each command ten times to specify up to ten IP addresses.

NOTE
You cannot restrict remote management access using the Web Management Interface.

Restricting Telnet access to a specific IP address

To allow Telnet access to the PowerConnect router only to the host with IP address 209.157.22.39, enter the following command.

```
NetIron(config)# telnet client 209.157.22.39
```

Syntax: [no] telnet client <*ip-addr*>

Restricting SSH access to a specific IP address

To allow SSH access to the PowerConnect router only to the host with IP address 209.157.22.39, enter the following command.

```
NetIron(config)# ip ssh client 209.157.22.39
```

Syntax: [no] ip ssh client <*ip-addr*>

Restricting SNMP access to a specific IP address

To allow SNMP access to the PowerConnect router only to the host with IP address 209.157.22.14, enter the following command.

```
NetIron(config)# snmp-client 209.157.22.14
```

Syntax: [no] snmp-client <*ip-addr*>

Restricting all remote management access to a specific IP address

To allow Telnet, and SNMP management access to the PowerConnect router only to the host with IP address 209.157.22.69, you can enter three separate commands (one for each access type) or you can enter the following command.

```
NetIron(config)# all-client 209.157.22.69
```

Syntax: [no] all-client <*ip-addr*>

Defining the Telnet idle time

You can define how many minutes a Telnet session can remain idle before it is timed out. An idle Telnet session is a session that is still sending TCP ACKs in response to keepalive messages from the device, but is not being used to send data.

To configure the idle time for a Telnet session, use the following command.

```
NetIron(config)# telnet timeout 120
```

Syntax: [no] telnet timeout <*0 – 240*>

Possible values: 0 – 240 minutes

Default value: 0 minutes (no timeout)

NOTE
The standard for the idle-timeout RADIUS attribute is for it to be implemented in seconds as opposed to the minutes that the PowerConnect router uses. If this attribute is used for setting idle time instead of this configuration, the value from the idle-timeout RADIUS attribute will be converted from seconds to minutes and truncated to the nearest minute.

Specifying the maximum login attempts for Telnet access

If you are connecting to the PowerConnect router using Telnet, the device prompts you for a username and password. By default, you have up to 4 chances to enter a correct username and password. If you do not enter a correct username or password after 4 attempts, the PowerConnect router disconnects the Telnet session.

You can specify the number of attempts a Telnet user has to enter a correct username and password before the PowerConnect router disconnects the Telnet session. For example, to allow a Telnet user up to 5 chances to enter a correct username and password, enter the following command.

```
NetIron(config)# telnet login-retries 5
```

Syntax: [no] telnet login-retries <*number*>

You can specify from 0 – 5 attempts. The default is 4 attempts.

Restricting remote access to the device to specific VLAN IDs

You can restrict management access to a PowerConnect router to ports within a specific port-based VLAN. VLAN-based access control applies to the following access methods:

- Telnet access
- SNMP access
- TFTP access

By default, access is allowed for all the methods listed above on all ports. Once you configure security for a given access method based on VLAN ID, access to the device using that method is restricted to only the ports within the specified VLAN.

VLAN-based access control works in conjunction with other access control methods. For example, suppose you configure an ACL to permit Telnet access only to specific client IP addresses, and you also configure VLAN-based access control for Telnet access. In this case, the only Telnet clients that can access the device are clients that have one of the IP addresses permitted by the ACL ***and*** are connected to a port that is in a permitted VLAN. Clients who have a permitted IP address but are connected to a port in a VLAN that is not permitted still cannot access the device through Telnet.

Restricting Telnet access to a specific VLAN

To allow Telnet access only to clients in a specific VLAN, enter a command such as the following.

```
NetIron(config)# telnet server enable vlan 10
```

The command configures the device to allow Telnet management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

Syntax: [no] telnet server enable vlan <*vlan-id*>

Restricting SNMP access to a specific VLAN

To allow SNMP access only to clients in a specific VLAN, enter a command such as the following.

```
NetIron(config)# snmp-server enable vlan 40
```

The command configures the device to allow SNMP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] snmp-server enable vlan <*vlan-id*>

Restricting TFTP access to a specific VLAN

To allow TFTP access only to clients in a specific VLAN, enter a command such as the following.

```
NetIron(config)# tftp client enable vlan 40
```

The command in this example configures the device to allow TFTP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] tftp client enable vlan <*vlan-id*>

Enabling specific access methods

You can specifically enable the following access methods:

- Telnet access
- SNMP access

NOTE
If you do not enable Telnet access, you can access the CLI using a serial connection to the management module. If you do not enable SNMP access, you will not be able to use SNMP management applications.

Enabling Telnet access

Telnet access is disabled by default. You can use a Telnet client to access the CLI on the device over the network.

To enable Telnet operation, enter the following command.

```
NetIron(config)# telnet-server
```

If you do not plan to use the CLI over the network and want to disable Telnet access to prevent others from establishing CLI sessions with the device, enter the following command.

```
NetIron(config)# no telnet-server
```

Syntax: [no] telnet-server

Enabling SNMP access

SNMP is disabled by default on the PowerConnect devices. SNMP is required if you want to manage a PowerConnect router using SNMP Network Manager.

To enable SNMP management of the device.

```
NetIron(config)#snmp-server
```

To later disable SNMP management of the device.

```
NetIron(config)#no snmp-server
```

Syntax: [no] snmp-server

Setting passwords

Passwords can be used to secure the following access methods:

- Telnet access can be secured by setting a Telnet password. Refer to "Setting a Telnet password" on page 27.
- Access to the Privileged EXEC and CONFIG levels of the CLI can be secured by setting passwords for management privilege levels. Refer to "Setting passwords for management privilege levels" on page 27.

This section also provides procedures for enhancing management privilege levels, recovering from a lost password, and disabling password encryption.

NOTE
You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account a management privilege level. Refer to "Setting up local user accounts" on page 31.

Setting a Telnet password

By default, the device does not require a user name or password when you log in to the CLI using Telnet.

To set the password "letmein" for Telnet access to the CLI, enter the following command at the global CONFIG level.

```
NetIron(config)# enable telnet password letmein
```

Syntax: [no] enable telnet password <*string*>

Suppressing Telnet connection rejection messages

By default, if a PowerConnect router denies Telnet management access to the device, the software sends a message to the denied Telnet client. You can optionally suppress the rejection message. When you enable the option, a denied Telnet client does not receive a message from the PowerConnect router. Instead, the denied client simply does not gain access.

To suppress the connection rejection message sent by the device to a denied Telnet client, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# telnet server suppress-reject-message
```

Syntax: [no] telnet server suppress-reject-message

Setting passwords for management privilege levels

You can set one password for each of the following management privilege levels:

- **Super User level** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords.
- **Port Configuration level** – Allows read-and-write access for specific ports but not for global (system-wide) parameters.
- **Read Only level** – Allows access to the Privileged EXEC mode and CONFIG mode of the CLI but only with read access.

You can assign a password to each management privilege level. You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account to one of the three privilege levels. Refer to "Setting up local user accounts" on page 31.

NOTE
You must use the CLI to assign a password for management privilege levels. You cannot assign a password using the Web Management Interface.

If you configure user accounts in addition to privilege level passwords, the device will validate a user's access attempt using one or both methods (local user account or privilege level password), depending on the order you specify in the authentication-method lists. Refer to "Configuring authentication-method lists" on page 73.

Follow the steps listed below to set passwords for management privilege levels.

1. At the opening CLI prompt, enter the following command to change to the Privileged level of the EXEC mode.

```
NetIron> enable
NetIron#
```

2. Access the CONFIG level of the CLI by entering the following command.

```
NetIron# configure terminal
NetIron(config)#
```

3. Enter the following command to set the Super User level password.

```
NetIron(config)# enable super-user-password <text>
```

NOTE
You must set the Super User level password before you can set other types of passwords. The Super User level password can be an alphanumeric string, but cannot begin with a number.

4. Enter the following commands to set the Port Configuration level and Read Only level passwords.

```
NetIron(config)# enable port-config-password <text>
NetIron(config)# enable read-only-password <text>
```

Syntax: **enable super-user-password** <*text*>

Syntax: **enable port-config-password** <*text*>

Syntax: **enable read-only-password** <*text*>

NOTE
If you forget your Super User level password, refer to "Recovering from a lost password" on page 29.

Augmenting management privilege levels

Each management privilege level provides access to specific areas of the CLI by default:

- Super User level provides access to all commands and displays.
- Port Configuration level gives access to the following:
  - The User EXEC and Privileged EXEC levels
  - The port-specific parts of the CONFIG level
  - All interface configuration levels
- Read Only level gives access to to the following:
  - The User EXEC and Privileged EXEC levels

You can grant additional access to a privilege level on an individual command basis. To grant the additional access, you specify the privilege level you are enhancing, the CLI level that contains the command, and the individual command.

NOTE
This feature applies only to management privilege levels on the CLI. You cannot augment management access levels for the Web Management Interface.

To enhance the Port Configuration privilege level so users also can enter IP commands at the global CONFIG level.

```
NetIron(config)# privilege configure level 4 ip
```

In this command, **configure** specifies that the enhanced access is for a command at the global CONFIG level of the CLI. The **level 4** parameter indicates that the enhanced access is for management privilege level 4 (Port Configuration). All users with Port Configuration privileges will have the enhanced access. The ip parameter indicates that the enhanced access is for the IP commands. Users who log in with valid Port Configuration level user names and passwords can enter commands that begin with "ip" at the global CONFIG level.

Syntax: [no] privilege <*cli-level*> **level** <*privilege-level*> <*command-string*>

The <*cli-level*> parameter specifies the CLI level and can be one of the following values:

- **exec** – EXEC level; for example, NetIron#
- **configure** – CONFIG level; for example, NetIron`(config)#`
- **interface** – Interface level; for example, NetIron`(config-if-e10000-6)#`
- **virtual-interface** – Virtual-interface level; for example, NetIron`(config-vif-6)#`
- **rip-router** – RIP router level; for example, NetIron`(config-rip-router)#`
- **ospf-router** – OSPF router level; for example, NetIron`(config-ospf-router)#`
- **bgp-router** – BGP4 router level; for example, NetIron`(config-bgp-router)#`
- **port-vlan** – Port-based VLAN level; for example, NetIron`(config-vlan)#`
- **protocol-vlan** – Protocol-based VLAN level
- **dot1x**
- **loopback-interface**
- **tunnel-interface**
- **vrrp-router**

The <*privilege-level*> indicates the number of the management privilege level you are augmenting. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The <*command-string*> parameter specifies the command you are allowing users with the specified privilege level to enter. To display a list of the commands at a CLI level, enter "?" at that level's command prompt.

Recovering from a lost password

Recovery from a lost password requires direct access to the serial port and a system reset.

NOTE
You can perform this procedure only from the CLI. This procedure should be performed with one primary management module installed. If both management modules are installed then the secondary will take over during the boot process, not allowing the password recovery process.

Follow the steps listed below to recover from a lost password.

1. Start a CLI session over the serial interface to the device.
2. Reboot the device.
3. At the initial boot prompt at system startup, enter **b** to enter the boot monitor mode.
4. Enter **no password** at the prompt. (You cannot abbreviate this command.) This command will cause the device to bypass the system password check.
5. Enter **boot system flash primary** at the prompt, and enter **y** when asked "Are you sure?".
6. After the console prompt reappears, assign a new password.

Displaying the SNMP community string

If you want to display the SNMP community string, enter the following commands.

```
NetIron(config)# enable password-display
NetIron(config)# show snmp server
```

The **enable password-display** command enables display of the community string, but only in the output of the **show snmp server** command. Display of the string is still encrypted in the startup configuration file and running configuration. Enter the command at the global CONFIG level of the CLI.

Disabling password encryption

When you configure a password, then save the configuration to the device's flash memory, the password is also saved to flash as part of the configuration file. By default, the passwords are encrypted so that the passwords cannot be observed by another user who displays the configuration file. Even if someone observes the file while it is being transmitted over TFTP, the password is encrypted.

If you want to remove the password encryption, you can disable encryption by entering the following command.

```
NetIron(config)# no service password-encryption
```

Syntax: [no] service password-encryption

Specifying a minimum password length

By default, the device imposes no minimum length on the Line (Telnet), Enable, or Local passwords. You can configure the device to require that Line, Enable, and Local passwords be at least a specified length.

For example, to specify that the Line, Enable, and Local passwords be at least 8 characters, enter the following command.

```
NetIron(config)# enable password-min-length 8
```

Syntax: [no] enable password-min-length <*number-of-characters*>

The <*number-of-characters*> can be from 1 – 48.

Setting up local user accounts

You can define up to 16 local user accounts on a PowerConnect router. User accounts regulate who can access the management functions in the CLI using the following methods:

- Telnet access
- SSH access
- Console access
- Web management access
- SNMP access

Local user accounts provide greater flexibility for controlling management access to the PowerConnect router than do management privilege level passwords and SNMP community strings of SNMP versions 1 and 2. You can continue to use the privilege level passwords and the SNMP community strings as additional means of access authentication. Alternatively, you can choose not to use local user accounts and instead continue to use only the privilege level passwords and SNMP community strings. Local user accounts are backward-compatible with configuration files that contain privilege level passwords. Refer to "Setting passwords for management privilege levels" on page 27.

If you configure local user accounts, you also need to configure an authentication-method list for Telnet access, Web management access, and SNMP access. Refer to "Configuring authentication-method lists" on page 73.

For each local user account, you specify a user name which can have up to 255 characters. You also can specify the following parameters:

- A password
- A management privilege level, which can be one of the following:
  - **Super User level** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only privilege level that allows you to configure passwords. This is the default.
  - **Port Configuration level** – Allows read-and-write access for specific ports but not for global (system-wide) parameters.
  - **Read Only level** – Allows access to the Privileged EXEC mode and CONFIG mode but only with read access.

Configuring a local user account

To configure a local user account, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# username wonka password willy
```

This command adds a local user account with the user name "wonka" and the password "willy". This account has the Super User privilege level; this user has full access to all configuration and display features.

NOTE
If you configure local user accounts, you must grant Super User level access to at least one account before you add accounts with other privilege levels. You need the Super User account to make further administrative changes.

```
NetIron(config)# username waldo privilege 5 password whereis
```

This command adds a user account for user name “waldo”, password “whereis”, with the Read Only privilege level. Waldo can look for information but cannot make configuration changes.

Syntax: [no] username <*user-string*> privilege <*privilege-level*> password | nopassword <*password-string*>

Enter up to 255 characters for <*user-string*>.

The **privilege** parameter specifies the privilege level for the account. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The default privilege level is **0**. If you want to assign Super User level access to the account, you can enter the command without **privilege 0**, as shown in the command example above.

The **password | nopassword** parameter indicates whether the user must enter a password. If you specify **password**, enter the string for the user's password.

NOTE
You must be logged on with Super User access (privilege level 0) to add user accounts or configure other access parameters.

To display user account information, enter the following command.

```
NetIron(config)# show users
```

Syntax: show users

Note about changing local user passwords

The PowerConnect router stores not only the current password configured for a local user, but the previous two passwords configured for the user as well. The local user's password cannot be changed to one of the stored passwords.

Consequently, if you change the password for a local user, you must select a password that is different from the current password, as well as different from the previous two passwords that had been configured for that user.

For example, say local user waldo originally had a password of "whereis", and the password was subsequently changed to “whois”, then later changed to “whyis”. If you change waldo's password again, you cannot change it to "whereis", "whois", or "whyis".

The current and previous passwords are stored in the device’s running configuration file in encrypted form.

Example

```
NetIron# show run
...
username waldo password 8 $1$Ro2..Ox0$udBu7pQT5XyuaXMUiUHy9. history
$1$eq...T62$IfpxIcxnDWX7CSVQKIodu. $1$QD3..2Q0$DYxgxCI64ZOSsYmSSaA28/
...
```

In the running configuration file, the user's previous two passwords are displayed in encrypted form following the **history** parameter.

Enhancements to username and password

The following rules are enabled by default:

- Users are now required to accept the message of the day.
- Users are locked out (disabled) if they fail to login after three attempts. This feature is automatically enabled.

The following rules are disabled by default:

- Enhanced user password combination requirements
- You can configure the system to store up to 15 previously configured passwords for each user.
- A password can now be set to expire.

Enabling enhanced user password combination requirements

Enhancements to the username and password have been added to provide added security. The regular password rules for username and password creation are still the default condition but the strict password rules can be enabled using the new command: "**enable strict-password-enforcement**" .

Regular password rules

The following rules apply to passwords unless the **enable strict-password-enforcement** command is executed:

- A minimum of one character is required to create a password.
- The last 3 passwords are stored in the CLI.
- No password expiration.
- Users were not locked out (disabled) after failed login attempts.

Strict password rules

The following rules have been implemented to enhance the password features in the PowerConnect devices.They apply to passwords if the **enable strict-password-enforcement** command is executed.

- Users are now required to accept the message of the day.
- Users are locked out (disabled) if they fail to login in three login attempts.

- The last 15 passwords are now stored in the CLI.
- A password can now be set to expire.
- Passwords are masked during password creation.
- Passwords must not share 4 or more concurrent characters with any other password configured on the router.
- You cannot configure a password that was previously configured.

When you create an enable and a user password, you must enter a minimum of eight characters containing the following combinations:

- At least two upper case characters
- At least two lower case characters
- At least two numeric characters
- At least two special character

NOTE
Password minimum and combination requirements are strictly enforced.

Configuring the strict password rules

Use the enable strict-password-enforcement command to enable the password security feature. Enter a command such as the following.

```
NetIron(config)# enable strict-password-enforcement
```

Syntax: [no] enable strict-password-enforcement

This feature is disabled by default.

The passwords must not share 4 or more concurrent characters with any other password configured on the router. If the user tries to create a password with 4 or more concurrent characters, the following error message will be returned.

```
Error - The substring <str> within the password has been used earlier, please
choose a different password.
```

Also, if the user tries to configure a password that was previously configured, the Local User Account configuration will not be allowed and the following message will be displayed.

```
This password was used earlier for same or different user, please choose a
different password."
```

When you create a password, the characters you type are masked.

Example : To create a password for the enable login.

```
NetIron(config)# user sandy password TesT12$%
```

Syntax: [no] user <*username*> password <*password*>

Example : To assign a password for a user account.

```
NetIron(config)# username sandy password [Enter]
Enter password: ********
```

Syntax: [no] username <*name*> password

Enter a password such as TesT12$! that contains the required character combination.

Requirement to accept the message of the day

If a message of the day (MOTD) is configured, a user will be required to press the "Enter" key before he or she can login. To enable this requirement, enter the command as shown.

```
NetIron(config)# banner motd require-enter-key
```

Login lockout

The CLI has been enhanced to provide up to three login attempts. If a user fails to login after three attempts, that user is locked out (disabled). To re-enable the user, do one of the following:

- Rebooting the PowerConnect router to re-enable all disabled users.
- Enable the user by entering the following command.

```
NetIron(config)# username sandy enable
```

Syntax: [no] username <*name*> **enable**

The <*name*> variable specifies the user whose access you want to enable.

There are no new CLI commands for this feature however the **enable strict-password-enforcement** command must be configured for this feature to be enabled.

Password history

The CLI has been enhanced to keep the last 15 passwords used by the user. A user will be prevented from changing their password to one that has already been used. This is for security purposes so that users do not use the same passwords multiple times.

There are no new CLI commands for this feature.

Setting passwords to expire

You can set a user password to expire. Once a password expires, the administrator must assign a new password to the user.

To configure a user password to expire, enter the following.

```
NetIron(config)# enable strict-password-enforcement
NetIron(config)# username sandy expires 20
```

Syntax: [no] username <*name*> **expires** <*days*>

The <*name*> variable specifies the user that the expiration time is applied to.

The <*days*> variable specifies the number of day before the password will expire. The following values can be used 1 – 365 days. The default is 90 days.

NOTE
The **enable strict-password-enforcement** command must be enabled before this command is configured. Otherwise, the following message will be displayed: `"Password expire time is enabled only if strict-password-enforcement is set".`

The **show user** command can be used to display the expiration date as shown in **bold** in the following.

```
NetIron(config)#show user Username Password Encrypt Priv Status Expire Time
================================================================================
= sandy $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled 0 enabled 20 days
```

Requirement to accept the message of the day

If a message of the day (MOTD) is configured, a user will be required to press the Enter key before he or she can login. MOTD is configured using the **banner motd** command.

There are no new CLI commands for this feature.

Enhanced login lockout

The CLI has been enhanced to provide up to three login attempts. If a user fails to login after three attempts, that user is locked out (disabled).

To re-enable a user that has been locked out, perform one of the following tasks:

- Reboot the PowerConnect to re-enable all disabled users.
- Enable the user by entering the following command.

  ```
  NetIron(config)# username sandy enable
  ```

Example

```
NetIron(config)# user sandy enable
PowerConnect(config)# show user
Username  Password                              Encrypt   Priv Status   Expire Time
=============================================================================

sandy     $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled   0   enabled  90 days
```

Syntax: [no] username <*name*> **enable**

Setting passwords to expire

You can set a user password to expire. Once a password expires, the administrator must assign a new password to the user. To configure a user password to expire, enter the following.

```
NetIron(config)# username sandy expires 20
```

Syntax: [no] username <*name*> **expires <days>**

Enter 1 – 365 for number of days. The default is 90 days.

Example

```
NetIron(config)# username sandy expires 20
NetIron(config)# show user
Username     Password                              Encrypt   Priv  Status   Expire
Time
================================================================================
=
sandy       $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled    0    enabled  20 days
```

Creating an encrypted all-numeric password

To create a password that is made up of all numeric values, use the command "**username** <*user-string*> **privilege** <*privilege-level*> **password** <*password-string*>." To allow backward compatibility with the **username** command, the new keyword **create-password** has been created and it is used as shown in the following.

```
NetIron# username customer1 create-password 9999
```

Syntax: [no] username <*user-string*> **create-password** <*password-string*>

The **create-password** option allows you to create a password with a numeric value in the <*password-string*> variable. The generated password will be encrypted. The **show running-config** command will display the password as shown.

```
username <user-string> 8 <encrypted-password>
```

Granting access by time of day

To configure an PowerConnect router to restrict access to a specified user to a specified time of day, use the following command.

```
NetIron(config)# username admin1 access-time 10:00:00 to 13:00:00
```

Syntax: [no] username <*user-string*> **access-time** <*hh:mm:ss*> **to** <*hh:mm:ss*>

The <*user-string*> variable specifies the user that you want to limit access time for.

The first instance of the <*hh:mm:ss*> variable specifies the start of the access time and the second instance of the <*hh:mm:ss*> variable specifies the end of the access time.

Configuring SSL security for the Web Management Interface

When enabled, the SSL protocol uses digital certificates and public-private key pairs to establish a secure connection to the PowerConnect router. Digital certificates serve to prove the identity of a connecting client, and public-private key pairs provide a means to encrypt data sent between the device and the client.

Configuring SSL for the Web Management Interface consists of the following tasks:

- Enabling the SSL server on the PowerConnect router
- Importing an RSA certificate and private key file from a client (optional)
- Generating a certificate

Enabling the SSL server on a PowerConnect router

To enable the SSL server on a PowerConnect router, enter the following command.

```
NetIron(config)# web-management https
```

Syntax: [no] web-management http | https

You can enable either the HTTP or HTTPs servers with this command. You can disable both the HTTP and HTTPs servers by entering the following command.

```
NetIron(config)# no web-management
```

Syntax: [no] web-management

Specifying a port for SSL communication

By default, SSL protocol exchanges occur on TCP port 443. You can optionally change the port number used for SSL communication.

For example, the following command causes the device to use TCP port 334 for SSL communication.

```
NetIron(config)# ip ssl port 334
```

Syntax: [no] ip ssl port <*port-number*>

The default port for SSL communication is 443.

Importing digital certificates and RSA private key files

To allow a client to communicate with the other PowerConnect router using an SSL connection, you configure a set of digital certificates and RSA public-private key pairs on the device. A digital certificate is used for identifying the server to the connecting client. It contains information about the issuing Certificate Authority, as well as a public key. You can either import digital certificates and private keys from a server, or you can allow the Dell device to create them.

If you want to allow the Dell device to create the digital certificates, refer to the next section, "Generating an SSL certificate". If you choose to import an RSA certificate and private key file from a client, you can use TFTP to transfer the files.

For example, to import a digital certificate using TFTP, enter a command such as the following.

```
NetIron(config)# ip ssl certificate-data-file tftp 192.168.9.210 certfile
```

Syntax: [no] ip ssl certificate-data-file tftp <*ip-addr*> <*certificate-filename*>

NOTE
If you import a digital certificate from a client, it can be no larger than 2048 bytes.

To import an RSA private key from a client using TFTP, enter a command such as the following.

```
NetIron(config)# ip ssl private-key-file tftp 192.168.9.210 keyfile
```

Syntax: [no] ip ssl private-key-file tftp <*ip-addr*> <*key-filename*>

The <*ip-addr*> is the IP address of a TFTP server that contains the digital certificate or private key.

Generating an SSL certificate

If you did not already import a digital certificate from a client, the device can create a default certificate. To do this, enter the following command.

```
NetIron(config)# crypto-ssl certificate generate
```

Syntax: [no] crypto-ssl certificate generate

Deleting the SSL certificate

To delete the SSL certificate, enter the following command.

```
NetIron(config)# crypto-ssl certificate zeroize
```

Syntax: [no] crypto-ssl certificate zeroize

Configuring TACACS or TACACS+ security

You can use the security protocol Terminal Access Controller Access Control System (TACACS) or TACACS+ to authenticate the following kinds of access to the PowerConnect devices:

- Telnet access
- SSH access
- Console access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

NOTE
You cannot authenticate SNMP access to a PowerConnect router using TACACS or TACACS+.

The TACACS and TACACS+ protocols define how authentication, authorization, and accounting information is sent between a PowerConnect router and an authentication database on a TACACS or TACACS+ server. TACACS or TACACS+ services are maintained in a database, typically on a UNIX workstation or PC with a TACACS or TACACS+ server running.

How TACACS+ differs from TACACS

TACACS is a simple UDP-based access control protocol originally developed by BBN for MILNET. TACACS+ is an enhancement to TACACS and uses TCP to ensure reliable delivery.

TACACS+ is an enhancement to the TACACS security protocol. TACACS+ improves on TACACS by separating the functions of authentication, authorization, and accounting (AAA) and by encrypting all traffic between the PowerConnect router and the TACACS+ server. TACACS+ allows for arbitrary length and content authentication exchanges, which allow any authentication mechanism to be utilized with the PowerConnect router. TACACS+ is extensible to provide for site customization and future development features. The protocol allows the PowerConnect router to request very precise access control and allows the TACACS+ server to respond to each component of that request.

NOTE
TACACS+ provides for authentication, authorization, and accounting, but an implementation or configuration is not required to employ all three.

TACACS or TACACS+ authentication, authorization, and accounting

When you configure a PowerConnect router to use a TACACS or TACACS+ server for authentication, the device prompts users who are trying to access the CLI for a user name and password, then verifies the password with the TACACS or TACACS+ server.

If you are using TACACS+, it is recommended that you also configure ***authorization***, in which the PowerConnect router consults a TACACS+ server to determine which management privilege level (and which associated set of commands) an authenticated user is allowed to use. You can also optionally configure ***accounting***, which causes the PowerConnect router to log information on the TACACS+ server when specified events occur on the device.

NOTE
By default, a user logging into the device through Telnet or SSH would first enter the User EXEC level. The user can enter the **enable** command to get to the Privileged EXEC level.

NOTE
A user that is successfully authenticated can be automatically placed at the Privileged EXEC level after login. Refer to "Entering privileged EXEC mode after a console, Telnet or SSH login" on page 48.

TACACS authentication

NOTE
Also, multiple challenges are supported for TACACS+ login authentication.

The following events occur when TACACS authentication takes place.

1. A user attempts to gain access to the PowerConnect router by doing one of the following:
   - Logging into the device using console, Telnet, SSH, or the Web Management Interface
   - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The PowerConnect router sends a request containing the username and password to the TACACS server.
5. The username and password are validated in the TACACS server's database.
6. If the password is valid, the user is authenticated.

TACACS+ authentication

The following events occur when TACACS+ authentication takes place.

1. A user attempts to gain access to the PowerConnect router by doing one of the following:
   - Logging into the device using console, telnet, SSH, or the Web Management Interface
   - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username.
3. The user enters a username.
4. The PowerConnect router obtains a password prompt from a TACACS+ server.
5. The user is prompted for a password.
6. The user enters a password.
7. The PowerConnect router sends the password to the TACACS+ server.

8. The password is validated in the TACACS+ server's database.
9. If the password is valid, the user is authenticated.

TACACS+ authorization

The PowerConnect devices support two kinds of TACACS+ authorization:

- Exec authorization determines a user's privilege level when they are authenticated.
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user.

The following events occur when TACACS+ exec authorization takes place.

1. A user logs into the PowerConnect router using console, Telnet or SSH
2. The user is authenticated.
3. The PowerConnect router consults the TACACS+ server to determine the privilege level of the user.
4. The TACACS+ server sends back a response containing an A-V (Attribute-Value) pair with the privilege level of the user.
5. The user is granted the specified privilege level.

The following events occur when TACACS+ command authorization takes place.

1. A Telnet, SSH, or console interface user previously authenticated by a TACACS+ server enters a command on the PowerConnect router.
2. The PowerConnect router looks at its configuration to see if the command is at a privilege level that requires TACACS+ command authorization.
3. If the command belongs to a privilege level that requires authorization, the PowerConnect router consults the TACACS+ server to see if the user is authorized to use the command.
4. If the user is authorized to use the command, the command is executed.

TACACS+ accounting

The following steps explain the working of TACACS+ accounting.

1. One of the following events occur on the PowerConnect router:
    - A user logs into the management interface using console, Telnet or SSH
    - A user enters a command for which accounting has been configured
    - A system event occurs, such as a reboot or reloading of the configuration file
2. The PowerConnect router checks its configuration to see if the event is one for which TACACS+ accounting is required.
3. If the event requires TACACS+ accounting, the PowerConnect router sends a TACACS+ Accounting Start packet to the TACACS+ accounting server, containing information about the event.
4. The TACACS+ accounting server acknowledges the Accounting Start packet.
5. The TACACS+ accounting server records information about the event.

6. When the event is concluded, the PowerConnect router sends an Accounting Stop packet to the TACACS+ accounting server.
7. The TACACS+ accounting server acknowledges the Accounting Stop packet.

AAA operations for TACACS or TACACS+

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a PowerConnect router that has TACACS or TACACS+ security configured.

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default <*method-list*>
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
	System accounting start (TACACS+): aaa accounting system default start-stop <*method-list*>
User logs in using console, Telnet, or SSH	Login authentication: aaa authentication login default <*method-list*>
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
	Exec accounting start (TACACS+): aaa accounting exec default <*method-list*> System accounting start (TACACS+): aaa accounting system default start-stop <*method-list*>
User logs into the Web Management Interface	Web authentication: aaa authentication web-server default <*method-list*>
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
User logs out of console, Telnet, or SSH session	Command accounting (TACACS+): aaa accounting commands <*privilege-level*> default start-stop <*method-list*> EXEC accounting stop (TACACS+): aaa accounting exec default start-stop <*method-list*>
User enters system commands (for example, **reload**, **boot system**)	Command authorization (TACACS+): aaa authorization commands <*privilege-level*> default <*method-list*>
	Command accounting (TACACS+): aaa accounting commands <*privilege-level*> default start-stop <*method-list*> System accounting stop (TACACS+): aaa accounting system default start-stop <*method-list*>

User action	Applicable AAA operations
User enters the command: [no] aaa accounting system default start-stop <*method-list*>	Command authorization (TACACS+): aaa authorization commands <*privilege-level*> default <*method-list*>
	Command accounting (TACACS+): aaa accounting commands <*privilege-level*> default start-stop <*method-list*> System accounting start (TACACS+): aaa accounting system default start-stop <*method-list*>
User enters other commands	Command authorization (TACACS+): aaa authorization commands <*privilege-level*> default <*method-list*>
	Command accounting (TACACS+): aaa accounting commands <*privilege-level*> default start-stop <*method-list*>

AAA Security for commands pasted Into the running configuration

If AAA security is enabled on a PowerConnect router, commands pasted into the running configuration are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running configuration, and AAA command authorization or accounting is configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running configuration. The server performing the AAA operations should be reachable when you paste the commands into the running configuration file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

TACACS or TACACS+ configuration considerations

Consider the following for configuring TACACS or TACACS+ servers:

- You must deploy at least one TACACS or TACACS+ server in your network.
- The PowerConnect router supports authentication using up to eight TACACS or TACACS+ servers. The device tries to use the servers in the order you add them to the device's configuration.
- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select TACACS+ as the primary authentication method for Telnet CLI access, but you cannot also select RADIUS authentication as a primary method for the same type of access. However, you can configure backup authentication methods for each access type.
- You can configure the Dell device to authenticate using a TACACS or TACACS+ server, not both.

TACACS configuration procedure

Use the following procedure for TACACS configurations.

1. Enable TACACS. "Enabling SNMP traps for TACACS" on page 44.
2. Identify TACACS servers. Refer to "Identifying the TACACS or TACACS+ servers" on page 44.
3. Set optional parameters. Refer to "Setting optional TACACS or TACACS+ parameters" on page 46.
4. Configure authentication-method lists. Refer to "Configuring authentication-method lists for TACACS or TACACS+" on page 47.

TACACS+ configuration procedure

Use the following procedure for TACACS+ configurations.

1. Enable TACACS. "Enabling SNMP traps for TACACS" on page 44
2. Identify TACACS+ servers. Refer to "Identifying the TACACS or TACACS+ servers" on page 44.
3. Set optional parameters. Refer to "Setting optional TACACS or TACACS+ parameters" on page 46.
4. Configure authentication-method lists. Refer to "Configuring authentication-method lists for TACACS or TACACS+" on page 47.
5. Optionally configure TACACS+ authorization. Refer to "Configuring TACACS+ authorization" on page 49.
6. Optionally configure TACACS+ accounting. Refer to "Configuring TACACS+ accounting" on page 52.

Enabling SNMP traps for TACACS

To enable SNMP traps for TACACS or TACACS+ on a PowerConnect router, you must execute the enable **snmp config-tacacs** command as shown in the following.

```
NetIron(config)# enable snmp config-tacacs
```

Syntax: [no] enable snmp [config-radius | config-tacacs]

The **config-radius** parameter specifies that traps will be enabled for RADIUS. Generation of Radius traps is disabled by default.

The **config-tacacs** parameter specifies that traps will be enabled for TACACS. Generation of TACACS traps is disabled by default.

Identifying the TACACS or TACACS+ servers

To use TACACS or TACACS+ servers to authenticate access to a PowerConnect router, you must identify the servers to the PowerConnect router.

For example, to identify three TACACS or TACACS+ servers, enter commands such as the following.

```
NetIron(config)# tacacs-server host 207.94.6.161
NetIron(config)# tacacs-server host 207.94.6.191
NetIron(config)# tacacs-server host 207.94.6.122
```

Syntax: [no] tacacs-server host <*ip-addr*> **|**<*hostname*> **[auth-port** <*number*>**]**

The <*ip-addr*> |<*hostname*> parameter specifies the IP address or host name of the server. You can enter up to eight **tacacs-server host** commands to specify up to eight different servers.

NOTE
To specify the server's host name instead of its IP address, you must first identify a DNS server using the **ip dns server-address** <*ip-addr*> command at the global CONFIG level.

If you add multiple TACACS or TACACS+ authentication servers to the PowerConnect router, the device tries to reach them in the order you add them. For example, if you add three servers in the following order, the software tries the servers in the same order.

1. 207.94.6.161
2. 207.94.6.191
3. 207.94.6.122

You can remove a TACACS or TACACS+ server by entering **no** followed by the **tacacs-server** command. For example, to remove 207.94.6.161, enter the following command.

```
NetIron(config)# no tacacs-server host 207.94.6.161
```

NOTE
If you erase a **tacacs-server** command (by entering "**no**" followed by the command), make sure you also erase the **aaa** commands that specify TACACS or TACACS+ as an authentication method. (Refer to "Configuring authentication-method lists for TACACS or TACACS+" on page 47.) Otherwise, when you exit from the CONFIG mode or from a Telnet session, the system continues to believe it is TACACS or TACACS+ enabled and you will not be able to access the system.

The **auth-port** parameter specifies the UDP (for TACACS) or TCP (for TACACS+) port number of the authentication port on the server. The default port number is 49.

Specifying different servers for individual AAA TACACS functions

In a TACACS+ configuration, you can designate a server to handle a specific AAA task. For example, you can designate one TACACS+ server to handle authorization and another TACACS+ server to handle accounting. You can set the TACACS+ key for each server.

To specify different TACACS+ servers for authentication, authorization, and accounting, enter a command such as the following.

```
NetIron(config)# tacacs-server host 1.2.3.4 auth-port 49 authentication-only key
abc
NetIron(config)# tacacs-server host 1.2.3.5 auth-port 49 authorization-only key
define
NetIron(config)# tacacs-server host 1.2.3.6 auth-port 49 accounting-only key ghi
```

Syntax: **[no] tacacs-server host** <*ip-addr*> **|** <*server-name*> **[auth-port** <*number*> **[authentication-only | authorization-only | accounting-only | default] [key** <*string*>**]]**

The **host** <*ip-addr*> | <*server-name*> parameter is either an IP address or an ASCII text string.

The **auth-port** <*number*> parameter is the Authentication port number; it is an optional parameter.

Enter **accounting-only** if the server is used only for TACACS accounting. Enter **authentication-only** if the server is used only for TACACS authentication. Enter **authorization-only** if the server is used only for TACAC authorization. Enterint the **default** parameter causes the server to be used for all AAA TACACS functions.

After authentication takes place, the server that performed the authentication is used for authorization, accounting or both. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until either a server that can perform the requested function is found, or every server in the configured list has been tried.

Enter **key** and configure a key for the server if an authentication key is to be used. By default, **key** is encrypted. If you want key to be in clear text, insert a **0** between **key** and <*string*>.

Example

```
NetIron(config)# tacacs-server host 1.2.3.5 auth-port 49 authorization-only key 0
report
```

The software adds a prefix to the authentication key string in the configuration. For example, the prefix "2" is added to the authorization key string in the following example.

```
tacacs-server host 1.2.3.6 auth-port 49 authorization-only key 2 $D?@d=8
```

The prefix can be one of the following:

- 0 = the key string is not encrypted and is in clear text
- 1 = the key string uses proprietary simple crytographic 2-way algorithm
- 2 = the key string uses proprietary base64 crytographic 2-way algorithm

Setting optional TACACS or TACACS+ parameters

You can set the following optional parameters in a TACACS or TACACS+ configuration:

- **TACACS+ key** – This parameter specifies the value that the Dell device sends to the TACACS+ server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the Dell device will resend an authentication request when the TACACS or TACACS+ server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- **Dead time** – This parameter specifies how long the Dell device waits for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 – 5 seconds. The default is 3 seconds.
- **Timeout** – This parameter specifies how many seconds the Dell device waits for a response from a TACACS or TACACS+ server before either retrying the authentication request, or determining that the TACACS or TACACS+ servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the TACACS+ key

The **key** parameter in the **tacacs-server** command is used to encrypt TACACS+ packets before they are sent over the network. The value for the **key** parameter on the PowerConnect router should match the one configured on the TACACS+ server. The key length can be from 1 – 64 characters and cannot include any space characters.

NOTE
The **tacacs-server key** command applies only to TACACS+ servers, not to TACACS servers. If you are configuring TACACS, do not configure a key on the TACACS server and do not enter a key on the PowerConnect router.

To specify a TACACS+ server key, enter the following command.

```
NetIron(config)# tacacs-server key rkwong
```

Syntax: [no] tacacs-server key [0 | 1] <*string*>

When you display the configuration of the PowerConnect router, the TACACS+ keys are encrypted.

Example

```
NetIron(config)# tacacs-server key 1 abc
NetIron(config)# write terminal
...
tacacs-server host 1.2.3.5 auth-port 49
tacacs key 1 $!2d
```

NOTE
Encryption of the TACACS+ keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the retransmission limit

The **retransmit** parameter specifies how many times the PowerConnect router will resend an authentication request when the TACACS or TACACS+ server does not respond. The retransmit limit can be from 1 – 5 times. The default is 3 times.

To set the TACACS or TACACS+ retransmit limit, enter the following command.

```
NetIron(config)# tacacs-server retransmit 5
```

Syntax: [no] tacacs-server retransmit <*number*>

Setting the timeout parameter

The **timeout** parameter specifies how many seconds the Dell device waits for a response from the TACACS or TACACS+ server before either retrying the authentication request, or determining that the TACACS or TACACS+ server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

```
NetIron(config)# tacacs-server timeout 5
```

Syntax: [no] tacacs-server timeout <*number*>

Configuring authentication-method lists for TACACS or TACACS+

You can use TACACS or TACACS+ to authenticate console, Telnet, or SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring TACACS or TACACS+ authentication, you create authentication-method lists specifically for these access methods, specifying TACACS or TACACS+ as the primary authentication method.

Within the authentication-method list, TACACS or TACACS+ is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If TACACS or TACACS+ authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list. If a TACACS or TACACS+ server responds with a reject for a user, the system does not try the backup authentication methods.

When you configure authentication-method lists for TACACS or TACACS+ authentication, you must create a separate authentication-method list for Telnet or SSH CLI access, and for access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication-method list that specifies TACACS or TACACS+ as the primary authentication method for securing Telnet or SSH access to the CLI.

```
NetIron(config)# enable telnet authentication
NetIron(config)# aaa authentication login default tacacs+ local
```

NOTE
To enable AAA support for commands entered at the console you must follow the procedure described in "Configuring AAA on the console" on page 72.

The commands above cause TACACS or TACACS+ to be the primary authentication method for securing Telnet or SSH access to the CLI. If TACACS or TACACS+ authentication fails due to an error with the server, authentication is performed using local user accounts instead.

To create an authentication-method list that specifies TACACS or TACACS+ as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI.

```
NetIron(config)# aaa authentication enable default tacacs+ local none
```

The command above causes TACACS or TACACS+ to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If TACACS or TACACS+ authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

For information on the command syntax, refer to page page 75 under "Examples of authentication-method lists".

NOTE
For examples of how to define authentication-method lists for types of authentication other than TACACS or TACACS+, refer to "Configuring authentication-method lists" on page 73.

Entering privileged EXEC mode after a console, Telnet or SSH login

By default, a user enters User EXEC mode after a successful login through console, Telnet or SSH. Optionally, you can configure the device so that a user enters Privileged EXEC mode after a console, Telnet or SSH login. To do this, use the following command.

```
NetIron(config)# aaa authentication login privilege-mode
```

Syntax: [no] aaa authentication login privilege-mode

The user's privilege level is based on the privilege level granted during login.

Configuring enable authentication to use enable password on TACACS+

TACACS+ server allows a common enable password to be configured on the TACACS+ server. To allow a user to authenticate against that enable password, instead of the login password, use this command.

```
NetIron(config)# aaa authentication enable implicit-user
```

Syntax: [no] aaa authentication enable implicit-user

Telnet or SSH prompts when the TACACS+ server is unavailable

When TACACS+ is the first method in the authentication method list, the device displays the login prompt received from the TACACS+ server. If a user attempts to login through Telnet or SSH, but none of the configured TACACS+ servers are available, the following takes place:

- If the next method in the authentication method list is “enable”, the login prompt is skipped, and the user is prompted for the Enable password (that is, the password configured with the **enable super-user-password** command).
- If the next method in the authentication method list is “line”, the login prompt is skipped, and the user is prompted for the Line password (that is, the password configured with the **enable telnet password** command).

Configuring TACACS+ authorization

The PowerConnect router supports TACACS+ authorization for controlling access to management functions in the CLI. Two kinds of TACACS+ authorization are supported:

- Exec authorization determines a user’s privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

Configuring exec authorization

When TACACS+ exec authorization is performed, the PowerConnect router consults a TACACS+ server to determine the privilege level of the authenticated user.

To configure TACACS+ exec authorization on a PowerConnect router, enter the following command.

```
NetIron(config)# aaa authorization exec default tacacs+
```

Syntax: aaa authorization exec default tacacs+ | radius | none

If you specify **none**, or omit the **aaa authorization** exec command from the device’s configuration, no exec authorization is performed.

A user’s privilege level is obtained from the TACACS+ server in the “foundry-privlvl” A-V pair. If the **aaa authorization exec default tacacs** command exists in the configuration, the device assigns the user the privilege level specified by this A-V pair. If the command does not exist in the configuration, then the value in the “foundry-privlvl” A-V pair is ignored, and the user is granted Super User access.

NOTE
If the **aaa authorization exec default tacacs+** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the "foundry-privlvl" A-V pair received from the TACACS+ server. If the **aaa authorization exec default tacacs+** command does not exist in the configuration, then the value in the "foundry-privlvl" A-V pair is ignored, and the user is granted Super User access.

Also note that in order for the **aaa authorization exec default tacacs+** command to work, either the **aaa authentication enable default tacacs+** command, or the **aaa authentication login default tacacs+** command must also exist in the configuration.

Configuring an attribute-value pair on the TACACS+ server

During TACACS+ exec authorization, the Dell device expects the TACACS+ server to send a response containing an A-V (Attribute-Value) pair that specifies the privilege level of the user. When the PowerConnect router receives the response, it extracts an A-V pair configured for the Exec service and uses it to determine the user's privilege level.

To set a user's privilege level, you can configure the "foundry-privlvl" A-V pair for the Exec service on the TACACS+ server.

Example

```
user=bob {
   default service = permit
   member admin
   # Global password
   global = cleartext "cat"
   service = exec {
     foundry-privlvl = 0
        }
}
```

In this example, the A-V pair `foundry-privlvl = 0` grants the user full read-write access. The value in the foundry-privlvl A-V pair is an integer that indicates the privilege level of the user. Possible values are 0 for super-user level, 4 for port-config level, or 5 for read-only level. If a value other than 0, 4, or 5 is specified in the foundry-privlvl A-V pair, the default privilege level of 5 (read-only) is used. The foundry-privlvl A-V pair can also be embedded in the group configuration for the user. Refer to your TACACS+ documentation for the configuration syntax relevant to your server.

If the foundry-privlvl A-V pair is not present, the PowerConnect router extracts the last A-V pair configured for the Exec service that has a numeric value. The PowerConnect router uses this A-V pair to determine the user's privilege level.

Example

```
user=bob {
   default service = permit
   member admin
   # Global password
   global = cleartext "cat"
   service = exec {
     priv-lvl = 15
        }
}
```

The attribute name in the A-V pair is not significant; the PowerConnect router uses the last one that has a numeric value. However, the PowerConnect router interprets the value for a non-"foundry-privlvl" A-V pair differently than it does for a "foundry-privlvl" A-V pair. The following table lists how the PowerConnect router associates a value from a non-"foundry-privlvl" A-V pair with a Dell privilege level.

TABLE 5 Dell equivalents for non-"foundry-privlvl" A-V pair values

Value for non-"foundry-privlvl" A-V pair	Privilege level
15	0 (super-user)
From 14 – 1	4 (port-config)
Any other number or 0	5 (read-only)

In the example above, the A-V pair configured for the Exec service is `priv-lvl = 15`. The PowerConnect router uses the value in this A-V pair to set the user's privilege level to 0 (super-user), granting the user full read-write access.

In a configuration that has both a "foundry-privlvl" A-V pair and a non-"foudnry-privlvl" A-V pair for the Exec service, the non-"foundry-privlvl" A-V pair is ignored.

Example

```
user=bob {
   default service = permit
   member admin
   # Global password
   global = cleartext "cat"
   service = exec {
     foundry-privlvl = 4
     priv-lvl = 15
        }
}
```

In this example, the user would be granted a privilege level of 4 (port-config level). The `privlvl = 15` A-V pair is ignored by the PowerConnect router.

If the TACACS+ server has no A-V pair configured for the Exec service, the default privilege level of 5 (read-only) is grated to the user.

Configuring command authorization

When TACACS+ command authorization is enabled, the PowerConnect router consults a TACACS+ server to get authorization for commands entered by the user.

You enable TACACS+ command authorization by specifying a privilege level whose commands require authorization. For example, to configure the PowerConnect router to perform authorization for the commands available at the Super User privilege level (that is, all commands on the device), enter the following command.

```
NetIron(config)# aaa authorization commands 0 default tacacs+
```

Syntax: [no] aaa authorization commands <*privilege-level*> **default tacacs+ | radius | none**

The <*privilege-level*> parameter can be one of the following:

- **0** – Authorization is performed for commands available at the Super User level (all commands)
- **4** – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)

- **5** – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE
TACACS+ command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web Management Interface or SNMP.

TACACS+ command authorization is not performed for the following commands:

- **At all levels: exit**, **logout**, **end**, and **quit**.
- **At the Privileged EXEC level: enable** or **enable** <*text*>, where <*text*> is the password configured for the Super User privilege level.

If configured, command accounting is performed for these commands.

NOTE
To enable AAA support for commands entered at the console you must follow the procedure described in "Configuring AAA on the console" on page 72.

Configuring TACACS+ accounting

The PowerConnect router supports TACACS+ accounting for recording information about user activity and system events. When you configure TACACS+ accounting on a PowerConnect router, information is sent to a TACACS+ accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring TACACS+ accounting for Telnet or SSH (shell) access

To send an Accounting Start packet to the TACACS+ accounting server when an authenticated user establishes a Telnet or SSH session on the PowerConnect router, and an Accounting Stop packet when the user logs out.

```
NetIron(config)# aaa accounting exec default start-stop tacacs+
```

Syntax: [no] aaa accounting exec default start-stop radius | tacacs+ | none

Configuring TACACS+ accounting for CLI commands

You can configure TACACS+ accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure the PowerConnect router to perform TACACS+ accounting for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
NetIron(config)# aaa accounting commands 0 default start-stop tacacs+
```

An Accounting Start packet is sent to the TACACS+ accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

NOTE
If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: [no] aa accounting commands <*privilege-level*> **default start-stop radius | tacacs+ | none**

The <*privilege-level*> parameter can be one of the following:

- 0 – Records commands available at the Super User level (all commands)
- 4 – Records commands available at the Port Configuration level (port-config and read-only commands)
- 5 – Records commands available at the Read Only level (read-only commands)

Configuring TACACS+ accounting for system events

You can configure TACACS+ accounting to record when system events occur on the PowerConnect router. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the TACACS+ accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed.

```
NetIron(config)# aaa accounting system default start-stop tacacs+
```

Syntax: [no] aaa accounting system default start-stop radius | tacacs+ | none

Configuring an interface as the source for all TACACS or TACACS+ packets

You can designate the lowest-numbered IP address configured an Ethernet port, loopback interface, or virtual interface as the source IP address for all TACACS or TACACS+ packets from the PowerConnect router. Identifying a single source IP address for TACACS or TACACS+ packets provides the following benefits:

- If your TACACS or TACACS+ server is configured to accept packets only from specific links or IP addresses, you can use this feature to simplify configuration of the TACACS or TACACS+ server by configuring the Dell device to always send the TACACS or TACACS+ packets from the same link or source address.
- If you specify a loopback interface as the single source for TACACS or TACACS+ packets, TACACS or TACACS+ servers can receive the packets regardless of the states of individual links. Thus, if a link to the TACACS or TACACS+ server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, TACACS or TACACS+, and RADIUS packets. You can configure a source interface for one or more of these types of packets.

To specify an Ethernet, loopback, or virtual interface as the source for all TACACS or TACACS+ packets from the device, use the following CLI method. The software uses the lowest-numbered IP address configured on the port or interface as the source IP address for TACACS or TACACS+ packets originated by the device.

To specify the lowest-numbered IP address configured on a virtual interface as the device source for all TACACS or TACACS+ packets, enter commands such as the following.

```
NetIron(config)# int ve 1
NetIron(config-vif-1)# ip address 10.0.0.3/24
NetIron(config-vif-1)# exit
NetIron(config)# ip tacacs source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all TACACS or TACACS+ packets from the PowerConnect router.

Syntax: [no]ip tacacs source-interface ethernet <*portnum*> **| loopback** <*num*> **| ve** <*num*>

The <*num*> parameter is a loopback interface or virtual interface number. If you specify an Ethernet, the <*portnum*> is the port's number (including the slot number, if you are configuring a device).

Displaying TACACS or TACACS+ statistics and configuration information

The **show aaa** command displays information about all TACACS+ and RADIUS servers identified on the device.

Example

```
NetIron# show aaa
Tacacs+ key: powerconnect
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server:  207.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

Syntax: show aaa

The following table describes the TACACS or TACACS+ information displayed by the **show aaa** command.

TABLE 6 Output of the show aaa command for TACACS or TACACS+

Field	Description
Tacacs+ key	The setting configured with the **tacacs-server key** command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Tacacs+ retries	The setting configured with the **tacacs-server retransmit** command.
Tacacs+ timeout	The setting configured with the **tacacs-server timeout** command.

TABLE 6 Output of the show aaa command for TACACS or TACACS+ (Continued)

Field	Description
Tacacs+ dead-time	The setting configured with the **tacacs-server dead-time** command.
Tacacs+ Server	For each TACACS or TACACS+ server, the IP address, port, and the following statistics are displayed: **opens** – Number of times the port was opened for communication with the server **closes** – Number of times the port was closed normally **timeouts** – Number of times port was closed due to a timeout **errors** – Number of times an error occurred while opening the port **packets in** – Number of packets received from the server **packets out** – Number of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.

The **show web** command displays the privilege level of Web Management Interface users.

Example

```
NetIron#show web
User                              Privilege       IP address
set                                      0       192.168.1.234
```

Syntax: show web

Configuring RADIUS security

You can use a Remote Authentication Dial In User Service (RADIUS) server to secure the following types of access to the PowerConnect devices:

- Telnet access
- SSH access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

NOTE
The PowerConnect devices do not support RADIUS security for SNMP access.

RADIUS authentication, authorization, and accounting

When RADIUS authentication is implemented, the PowerConnect router consults a RADIUS server to verify usernames and passwords. Optionally, you can configure RADIUS authorization, in which the PowerConnect router consults a list of commands supplied by the RADIUS server to determine whether a user can execute a command that has been entered. You can also configure RADIUS accounting, which causes the PowerConnect router to log information on a RADIUS accounting server when specified events occur on the device.

NOTE
By default, a user logging into the device through Telnet or SSH first enters the User EXEC level. The user can then enter the **enable** command to get to the Privileged EXEC level.

NOTE
A user that is successfully authenticated can be automatically placed at the Privileged EXEC level after login. Refer to "Entering privileged EXEC mode after a Telnet or SSH login" on page 66.

RADIUS authentication

The following events occur when RADIUS authentication takes place.

1. A user triggers RADIUS authentication by doing one of the following:
   - Logging in to the device using Telnet or SSH
   - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The PowerConnect device sends a RADIUS Access-Request packet containing the username and password to the RADIUS server.
5. The RADIUS server validates the PowerConnect device using a shared secret (the RADIUS key).
6. The RADIUS server looks up the username in its database.
7. If the username is found in the database, the RADIUS server validates the password.
8. If the password is valid, then:
   a. If the RADIUS server is configured to use multi-factor authentication, it may send an Access-Challenge packet to the PowerConnect device. If so, the user may be asked for additional input (for example, an RSA SecurID PIN or RSA SecurID next tokencode) which the PowerConnect device will forward to the RADIUS server. If the additional input is valid, then the process moves to the next step.
   b. If the RADIUS server is configured to use single-factor authentication, then the process moves immediately to the next step.
9. The RADIUS server sends an Access-Accept packet to the PowerConnect device, authenticating the user. Within the Access-Accept packet are three Dell vendor-specific attributes that indicate:
   - The privilege level of the user
   - A list of commands
   - Whether the user is allowed or denied usage of the commands in the list

   The last two attributes are used with RADIUS authorization, if configured.
10. The user is authenticated, and the information supplied in the Access-Accept packet for the user is stored on the PowerConnect router. The user is granted the specified privilege level. If you configure RADIUS authorization, the user is allowed or denied usage of the commands in the list.

Multi-factor RADIUS authentication

The PowerConnect device supports multi-factor authentication (for example, RSA SecurID) via a RADIUS server. For access by Telnet, no further configuration is needed on the PowerConnect device to enable multi-factor RADIUS authentication. For multi-factor authentication when using SSH, the PowerConnect device must be configured for interactive authentication using the following command.

```
NetIron(config)# ip ssh interactive-authentication yes
```

Syntax: ip ssh interactive-authentication {**no** | **yes**}

The default is **yes** (interactive authentication is supported by default).

A sample interactive authentication session (with RSA SecurID) is shown below.

```
Telnet_DMT_MLXe_16k - 08-25-2010  -- 11:20:18  Session Log Start --
10.20.179.55|Telnet - 08-25-2010  -- 11:20:18

Telnet - 08-25-2010  -- 11:20:18  This is the message of the day
Telnet - 08-25-2010  -- 11:20:18
Telnet - 08-25-2010  -- 11:20:18  User Access Verification
Telnet - 08-25-2010  -- 11:20:18
Telnet - 08-25-2010  -- 11:20:38  Please Enter Login Name: pbikram3
Telnet - 08-25-2010  -- 11:20:58  Please Enter Password:
<enter-token-code-for-user-here>
Telnet - 08-25-2010  -- 11:21:01
Telnet - 08-25-2010  -- 11:21:06  Enter a new PIN having from 4 to 8 alphanumeric
characters:<new-pin>
Telnet - 08-25-2010  -- 11:21:07
Telnet - 08-25-2010  -- 11:21:10  Please re-enter new PIN:<new-pin>
Telnet - 08-25-2010  -- 11:21:12
Telnet - 08-25-2010  -- 11:21:12  PIN Accepted.
Telnet - 08-25-2010  -- 11:21:12  Wait for the token code to change,
Telnet - 08-25-2010  -- 11:21:36  then enter the new
passcode:<new-pin>+<enter-token-code-for-user-here>
Telnet - 08-25-2010  -- 11:21:38
Telnet - 08-25-2010  -- 11:21:38  User login successful.
Telnet - 08-25-2010  -- 11:21:38


Telnet - 08-25-2010  -- 11:21:55  Session Log End --10.20.179.55|Telnet -
08-25-2010  -- 11:21:55
```

RADIUS authorization

The following events occur when RADIUS authorization takes place.

1. A user previously authenticated by a RADIUS server enters a command on the PowerConnect router.
2. The PowerConnect router looks at its configuration to see if the command is at a privilege level that requires RADIUS command authorization.
3. If the command belongs to a privilege level that requires authorization, the PowerConnect router looks at the list of commands delivered to it in the RADIUS Access-Accept packet when the user was authenticated. (Along with the command list, an attribute was sent that specifies whether the user is permitted or denied usage of the commands in the list.)

NOTE
After RADIUS authentication takes place, the command list resides on the PowerConnect router. The RADIUS server is not consulted again once the user has been authenticated. This means that any changes made to the user's command list on the RADIUS server are not reflected until the next time the user is authenticated by the RADIUS server, and the new command list is sent to the PowerConnect router.

4. If the command list indicates that the user is authorized to use the command, the command is executed.

RADIUS accounting

The following steps explain the working of RADIUS accounting.

1. One of the following events occur on a PowerConnect router:
   - A user logs into the management interface using Telnet or SSH
   - A user enters a command for which accounting has been configured
   - A system event occurs, such as a reboot or reloading of the configuration file
2. The PowerConnect router checks its configuration to see if the event is one for which RADIUS accounting is required.
3. If the event requires RADIUS accounting, the PowerConnect router sends a RADIUS Accounting Start packet to the RADIUS accounting server, containing information about the event.
4. The RADIUS accounting server acknowledges the Accounting Start packet.
5. The RADIUS accounting server records information about the event.
6. When the event is concluded, the PowerConnect router sends an Accounting Stop packet to the RADIUS accounting server.
7. The RADIUS accounting server acknowledges the Accounting Stop packet.

AAA operations for RADIUS

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a PowerConnect router that has RADIUS security configured.

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default <*method-list*>
	System accounting start: aaa accounting system default start-stop <*method-list*>
User logs in using Telnet or SSH	Login authentication: aaa authentication login default <*method-list*>
	EXEC accounting Start: aaa accounting exec default start-stop <*method-list*> System accounting Start: aaa accounting system default start-stop <*method-list*>
User logs into the Web Management Interface	Web authentication: aaa authentication web-server default <*method-list*>

User action	Applicable AAA operations
User logs out of Telnet or SSH session	Command authorization for **logout** command: aaa authorization commands <*privilege-level*> default <*method-list*>
	Command accounting: aaa accounting commands <*privilege-level*> default start-stop <*method-list*> EXEC accounting stop: aaa accounting exec default start-stop <*method-list*>
User enters system commands (for example, **reload**, **boot system**)	Command authorization: aaa authorization commands <*privilege-level*> default <*method-list*>
	Command accounting: aaa accounting commands <*privilege-level*> default start-stop <*method-list*> System accounting stop: aaa accounting system default start-stop <*method-list*>
User enters the command: [no] aaa accounting system default start-stop <*method-list*>	Command authorization: aaa authorization commands <*privilege-level*> default <*method-list*>
	Command accounting: aaa accounting commands <*privilege-level*> default start-stop <*method-list*> System accounting start: aaa accounting system default start-stop <*method-list*>
User enters other commands	Command authorization: aaa authorization commands <*privilege-level*> default <*method-list*>
	Command accounting: aaa accounting commands <*privilege-level*> default start-stop <*method-list*>

AAA security for commands pasted into the running configuration

If AAA security is enabled on the device, commands pasted into the running configuration are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running configuration, and AAA command authorization or accounting is configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running configuration. The server performing the AAA operations should be reachable when you paste the commands into the running configuration file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

NOTE
Since RADIUS command authorization relies on a list of commands received from the RADIUS server when authentication is performed, it is important that you use RADIUS authentication when you also use RADIUS command authorization.

RADIUS configuration considerations

Consider the following for configuring the RADIUS server:

- You must deploy at least one RADIUS server in your network.
- The PowerConnect router supports authentication using up to eight RADIUS servers. The device tries to use the servers in the order you add them to the device's configuration. If one RADIUS server is not responding, the device tries the next one in the list.
- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select RADIUS as the primary authentication method for Telnet CLI access, but you cannot also select TACACS+ authentication as the primary method for the same type of access. However, you can configure backup authentication methods for each access type.

RADIUS configuration procedure

Use the following procedure to configure a PowerConnect router for RADIUS.

1. Configure Dell vendor-specific attributes on the RADIUS server. Refer to "Configuring Dell-specific attributes on the RADIUS server" on page 60.
2. Enabling Radius. Refer to "Enabling SNMP traps for RADIUS" on page 62.
3. Identify the RADIUS server to the PowerConnect router. Refer to "Identifying the RADIUS server to the PowerConnect router" on page 63.
4. Set RADIUS parameters. Refer to "Setting RADIUS parameters" on page 64.
5. Configure authentication-method lists. Refer to "Configuring authentication-method lists for RADIUS" on page 65.
6. Optionally configure RADIUS authorization. Refer to "Configuring RADIUS authorization" on page 67.
7. Optionally configure RADIUS accounting. "Configuring RADIUS accounting" on page 68.

Configuring Dell-specific attributes on the RADIUS server

During the RADIUS authentication process, if a user supplies a valid username and password, the RADIUS server sends an Access-Accept packet to the PowerConnect, authenticating the user. Within the Access-Accept packet, the RADIUS server could send attribute "Vendor-Specific" (26) whose value could inform the PowerConnect on the runtime enviornment for this session. This section will detail all the vendor specific attributes defined by Dell.

TABLE 7 Vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Description
foundry-privilege-level	1	integer	Specifies the privilege level for the user. This attribute can be set to one of the following: **0** –**Super User level** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords. **4** – **Port Configuration level** – Allows read-and-write access for specific ports but not for global (system-wide) parameters. **5** – **Read Only level** – Allows access to the Privileged EXEC mode and CONFIG mode of the CLI but only with read access.
foundry-command-string	2	string	Specifies a list of CLI commands that are permitted or denied to the user when RADIUS authorization is configured. The commands are delimited by semi-colons (;). You can specify an asterisk (*) as a wildcard at the end of a command string. For example, the following command list specifies all **show** and **debug ip** commands, as well as the **write terminal** command: show *; debug ip *; write term*
powerconnect-command-exception-flag	3	integer	Specifies whether the commands indicated by the powerconnect-command-string attribute are permitted or denied to the user. This attribute can be set to one of the following: **0** – Permit execution of the commands indicated by powerconnect-command-string, deny all other commands. **1** – Deny execution of the commands indicated by powerconnect-command-string, permit all other commands.
foundry-INM-privilege	4	integer	Specifies the IronView Network Manager user privilege level. This attribute can take a value range from 0 to 15. In IronView Network Manager, this attribute value will be mapped to the preconfigured roles “AAA privilege level 0” through “AAA privilege level 15”. The admin user has to configure these roles with the appropriate sets of privileges in order for the AAA user to get the correct set of feature access.

TABLE 7 Vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Description
foundry-access-list	5	string	Specifies the access control list to be used for RADIUS authorization. Enter the access control list in the following format. type=string, value="ipacl.[e|s].[in|out] = [<acl-name>|<acl-number>] <separator> macfilter.in = [<acl-name>|<acl-number>] Where: • separator can be a space, newline, semicolon, comma, or null characater • ipacl.e is extended ACL; ipacl.s is standard ACL. **NOTE:** Outbound MAC filters are not supported, but outbound ACLs with 802.1X authentication is supported.
foundry-MAC-authent-needs-802x	6	integer	Specifies whether or not 802.1x authentication is required and enabled. **0** - Disabled **1** - Enabled
foundry-802.1x-valid-lookup	7	integer	Specifies if 802.1x lookup is enabled: **0** - Disabled **1** - Enabled
foundry-MAC-based-VLAN-QOS	8	integer	Specifies the priority for MAC-based VLAN QOS: **0** - qos_priority_0 **1** - qos_priority_1 **2** - qos_priority_2 **3** - qos_priority_3 **4** - qos_priority_4 **5** - qos_priority_5 **6** - qos_priority_6 **7** - qos_priority_7
foundry-INM-Role-AOR-List	9	string	Specifies the list of Roles and Area of Responsibility (AOR) that are allowed for an IronView Network Manager user. These values are mapped to IronView Network Manager Roles and AORs when the user logs in. For example, to configure an IronView Network Manager user to have “Administrator” and”'Report User” roles and “New York Region” and “Santa Clara Region” AORs, specify “'InmRoles=Administrator, Report User; InmAORs=New York Region, Santa Clara Region”. The keys “InmRoles” and “InmAORs” are delimited by semi colon (;) and the values for the keys are delimited by a comma (,). Refer to the *IronView Network Manager User Guide* for details.

Enabling SNMP traps for RADIUS

To enable SNMP traps for RADIUS on a PowerConnect router, you must execute the **enable snmp config-radius** command as shown in the following.

```
NetIron(config)# enable snmp config-radius
```

Syntax: [no] enable snmp [config-radius | config-tacacs]

The **config-radius** parameter specifies that traps will be enabled for RADIUS. Generation of Radius traps is disabled by default.

The **config-tacacs** parameter specifies that traps will be enabled for TACACS. Generation of TACACS traps is disabled by default.

Identifying the RADIUS server to the PowerConnect router

To use a RADIUS server to authenticate access to a PowerConnect router, you must identify the server to the PowerConnect router.

```
NetIron(config)# radius-server host 209.157.22.99
```

Syntax: [no] radius-server host <*ip-addr*> | <*server-name*> [auth-port <*number*> acct-port <*number*>]

The **host** <*ip-addr*> | <*server-name*> parameter is either an IP address or an ASCII text string.

The <*auth-port*> parameter is the Authentication port number; it is an optional parameter. The default is 1812.

The <*acct-port*> parameter is the Accounting port number; it is an optional parameter. The default is 1813.

Specifying different servers for individual AAA functions

In a RADIUS configuration, you can designate a server to handle a specific AAA task. For example, you can designate one RADIUS server to handle authorization and another RADIUS server to handle accounting. You can specify individual servers for authentication and accounting, but not for authorization. You can set the RADIUS key for each server.

To specify different RADIUS servers for authentication and accounting, enter a command such as the following.

```
NetIron(config)# radius-server host 1.2.3.4 auth-port 1812 acct-port 1813
authentication-only key abc
NetIron(config)# radius-server host 1.2.3.6 auth-port 1812 acct-port 1813
accounting-only key ghi
```

Syntax: [no] radius-server host <*ip-addr*> | <*server-name*> [auth-port <*number*> acct-port <*number*> [authentication-only | accounting-only | default] [key [0|1|2] <*string*> [dot1x]]]

The **host** <*ip-addr*> | <*server-name*> parameter is either an IP address or an ASCII text string.

The **auth-port** <*number*> parameter specifies what port to use for RADIUS authentication. The default is 1812.

The **acct-port** <*number*>parameter specifies what port to use for RADIUS accounting. The default is 1813.

Enter **accounting-only** if the server is used only for accounting. Enter **authentication-only** if the server is used only for authentication. Entering the **default** parameter causes the server to be used for all AAA RADIUS functions.

NOTE
To specify which RADIUS function(s) the server supports, you must first enter the authentication port and accounting port parameters.

After authentication takes place, the server that performed the authentication is used for authorization, accounting, or both. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until either a server that can perform the requested function is found, or every server in the configured list has been tried.

Enter **key** and configure a key for the server if an authentication key is to be used. By default, **key** is encrypted. If you want key to be in clear text, insert a **0** between **key** and <*string*>.

```
NetIron(config)# radius-server host 1.2.3.4 authentication-only key 0 abc
```

The software adds a prefix to the authentication key in the configuration. For example, the prefix "2" is added to the key string in the example below.

```
radius-server host 1.2.3.6 auth-port 1812 acct-port 1813 default key 2 $D?@d=8
```

The prefix can be one of the following:

- **0** = the key string is not encrypted and is in clear text
- **1** = the key string uses proprietary simple crytographic 2-way algorithm
- **2** = the key string uses proprietary base64 crytographic 2-way algorithm

Setting RADIUS parameters

You can set the following parameters in a RADIUS configuration:

- **RADIUS key** – This parameter specifies the value that the PowerConnect router sends to the RADIUS server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the PowerConnect router will resend an authentication request when the RADIUS server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- **Timeout** – This parameter specifies how many seconds the PowerConnect router waits for a response from a RADIUS server before either retrying the authentication request, or determining that the RADIUS servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the RADIUS key

The **key** parameter in the **radius-server** command is used to encrypt RADIUS packets before they are sent over the network. The value for the **key** parameter on the PowerConnect router should match the one configured on the RADIUS server. The key length can be from 1 – 64 characters and cannot include any space characters.

To specify a RADIUS server key, enter a command such as the following.

```
NetIron(config)# radius-server key mirabeau
```

Syntax: [no] radius-server key [0 | 1] <*string*>

When you display the configuration of the PowerConnect router, the RADIUS key is encrypted.

Example

```
NetIron(config)# radius-server key 1 abc
NetIron(config)# write terminal
...
radius-server host 1.2.3.5
radius key 1 $!2d
```

NOTE
Encryption of the RADIUS keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the retransmission limit

The **retransmit** parameter specifies the maximum number of retransmission attempts. When an authentication request times out, the software will retransmit the request up to the maximum number of retransmissions configured. The default retransmit value is 3 retries. The range of retransmit values is from 1 – 5.

To set the RADIUS retransmit limit, enter a command such as the following.

```
NetIron(config)# radius-server retransmit 5
```

Syntax: [no] radius-server retransmit <*number*>

Setting the timeout parameter

The **timeout** parameter specifies how many seconds the PowerConnect router waits for a response from the RADIUS server before either retrying the authentication request, or determining that the RADIUS server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

```
NetIron(config)# radius-server timeout 5
```

Syntax: [no] radius-server timeout <*number*>

Configuring authentication-method lists for RADIUS

You can use RADIUS to authenticate Telnet or SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring RADIUS authentication, you create authentication-method lists specifically for these access methods, specifying RADIUS as the primary authentication method.

Within the authentication-method list, RADIUS is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If RADIUS authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list.

When you configure authentication-method lists for RADIUS, you must create a separate authentication-method list for Telnet or SSH CLI access and for CLI access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing Telnet access to the CLI, enter the following command.

```
NetIron(config)# enable telnet authentication
NetIron(config)# aaa authentication login default radius local
```

The commands above cause RADIUS to be the primary authentication method for securing Telnet access to the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead.

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI, enter the following command.

```
NetIron(config)# aaa authentication enable default radius local none
```

The command above causes RADIUS to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

For information on the command syntax, refer to page page 75 under “Examples of authentication-method lists”.

NOTE
For examples of how to define authentication-method lists for types of authentication other than RADIUS, refer to “Configuring authentication-method lists” on page 73.

Entering privileged EXEC mode after a Telnet or SSH login

By default, a user enters User EXEC mode after a successful login through Telnet or SSH. You can configure the device so that a user enters Privileged EXEC mode after a Telnet or SSH login. To do this, use the following command.

```
NetIron(config)# aaa authentication login privilege-mode
```

Syntax: [no] aaa authentication login privilege-mode

The user’s privilege level is based on the privilege level granted during login.

Configuring enable authentication to prompt for password only

If Enable authentication is configured on the device, by default, a user is prompted for a username and password. when the user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI. You can configure the PowerConnect router to prompt only for a password. The device uses the username (up to 255 characters) entered at login, if one is available. If no username was entered at login, the device prompts for both username and password.

To configure the PowerConnect router to prompt only for a password when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI, enter the following command.

```
NetIron(config)# aaa authentication enable implicit-user
```

Syntax: [no] aaa authentication enable implicit-user

Configuring RADIUS authorization

The PowerConnect router supports RADIUS authorization for controlling access to management functions in the CLI. Two kinds of RADIUS authorization are supported:

- Exec authorization determines a user's privilege level when they are authenticated
- Command authorization consults a RADIUS server to get authorization for commands entered by the user

Configuring Exec authorization

NOTE
Before you configure RADIUS exec authorization on a PowerConnect router, make sure that the **aaa authentication enable default radius** command exists in the configuration.

When RADIUS exec authorization is performed, the PowerConnect router consults a RADIUS server to determine the privilege level of the authenticated user.

To configure RADIUS exec authorization on a PowerConnect router, enter the following command.

```
NetIron(config)# aaa authentication login default radius
NetIron(config)# aaa authorization exec default radius
```

Syntax: [no] aaa authorization exec default radius | none

If you specify **none**, or omit the **aaa authorization exec** command from the device's configuration, no exec authorization is performed.

NOTE
If the **aaa authorization exec default radius** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the powerconnect-privilege-level attribute received from the RADIUS server. If the **aaa authorization exec default radius** command does not exist in the configuration, then the value in the powerconnect-privilege-level attribute is ignored, and the user is granted Super User access.

For the **aaa authorization exec default radius** command to work, either the **aaa authentication login default radius** command, or the **aaa authentication enable default radius** command must also exist in the configuration.

Configuring command authorization

When RADIUS command authorization is enabled, the PowerConnect router consults the list of commands supplied by the RADIUS server during authentication to determine whether a user can execute a command he or she has entered.

You enable RADIUS command authorization by specifying a privilege level whose commands require authorization. For example, to configure the PowerConnect router to perform authorization for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
NetIron(config)# aaa authorization commands 0 default radius
```

Syntax: [no] aaa authorization commands <*privilege-level*> **default radius | tacacs+ | none**

The <*privilege-level*> parameter can be one of the following:

- 0 – Authorization is performed (that is, the PowerConnect router looks at the command list) for commands available at the Super User level (all commands)
- 4 – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- 5 – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE
RADIUS command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web Management Interface or SNMP.

NOTE
Since RADIUS command authorization relies on the command list supplied by the RADIUS server during authentication, you cannot perform RADIUS authorization without RADIUS authentication.

Command authorization and accounting for console commands

The PowerConnect devices support command authorization and command accounting for CLI commands entered at the console. To configure the device to perform command authorization and command accounting for console commands, enter the following.

```
NetIron(config)# enable aaa console
```

Syntax: [no] enable aaa console

CAUTION
If you have previously configured the device to perform command authorization using a RADIUS server, entering the enable aaa console command may prevent the execution of any subsequent commands entered on the console.

NOTE
This happens because RADIUS command authorization requires a list of allowable commands from the RADIUS server. This list is obtained during RADIUS authentication. For console sessions, RADIUS authentication is performed only if you have configured Enable authentication and specified RADIUS as the authentication method (for example, with the **aaa authentication enable default radius** command). If RADIUS authentication is never performed, the list of allowable commands is never obtained from the RADIUS server. Consequently, there would be no allowable commands on the console.

Configuring RADIUS accounting

The PowerConnect devices support RADIUS accounting for recording information about user activity and system events. When you configure RADIUS accounting on a PowerConnect router, information is sent to a RADIUS accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring RADIUS accounting for Telnet or SSH (shell) access

To send an Accounting Start packet to the RADIUS accounting server when an authenticated user establishes a Telnet or SSH session on the PowerConnect router, and an Accounting Stop packet when the user logs out, enter the following command.

```
NetIron(config)# aaa accounting exec default start-stop radius
```

Syntax: [no] aaa accounting exec default start-stop radius | tacacs+ | none

Configuring RADIUS accounting for CLI commands

You can configure RADIUS accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure a PowerConnect router to perform RADIUS accounting for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
NetIron(config)# aaa accounting commands 0 default start-stop radius
```

An Accounting Start packet is sent to the RADIUS accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

NOTE
If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: [no] aaa accounting commands <*privilege-level*> **default start-stop radius | tacacs | none**

The <*privilege-level*> parameter can be one of the following:

- 0 – Records commands available at the Super User level (all commands)
- 4 – Records commands available at the Port Configuration level (port-config and read-only commands)
- 5 – Records commands available at the Read Only level (read-only commands)

Configuring RADIUS accounting for system events

You can configure RADIUS accounting to record when system events occur on a PowerConnect router. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the RADIUS accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed.

```
NetIron(config)# aaa accounting system default start-stop radius
```

Syntax: [no] aaa accounting system default start-stop radius | tacacs+ | none

Configuring an interface as the source for all RADIUS packets

You can designate the lowest-numbered IP address configured an Ethernet port, loopback interface, or virtual interface as the source IP address for all RADIUS packets from the PowerConnect router. Identifying a single source IP address for RADIUS packets provides the following benefits:

- If your RADIUS server is configured to accept packets only from specific links or IP addresses, you can use this feature to simplify configuration of the RADIUS server by configuring the PowerConnect router to always send the RADIUS packets from the same link or source address.
- If you specify a loopback interface as the single source for RADIUS packets, RADIUS servers can receive the packets regardless of the states of individual links. Thus, if a link to the RADIUS server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, TACACS or TACACS+, and RADIUS packets. You can configure a source interface for one or more of these types of packets.

To specify an Ethernet or a loopback or virtual interface as the source for all RADIUS packets from the device, use the following CLI method. The software uses the lowest-numbered IP address configured on the port or interface as the source IP address for RADIUS packets originated by the device.

To specify the lowest-numbered IP address configured on a virtual interface as the device's source for all RADIUS packets, enter commands such as the following.

```
NetIron(config)# int ve 1
NetIron(config-vif-1)# ip address 10.0.0.3/24
NetIron(config-vif-1)# exit
NetIron(config)# ip radius source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all RADIUS packets from the PowerConnect router.

Syntax: [no] ip radius source-interface ethernet <*portnum*> **| loopback** <*num*> **| ve** <*num*>

The <*num*> parameter is a loopback interface or virtual interface number. If you specify an Ethernet port, the <*portnum*> is the port's number (including the slot number, if you are configuring a device).

Displaying RADIUS configuration information

The **show aaa** command displays information about all TACACS or TACACS+ and RADIUS servers identified on the device.

```
NetIron# show aaa
Tacacs+ key: powerconnect
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server:  207.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

Syntax: show aaa

The following table describes the RADIUS information displayed by the **show aaa** command.

TABLE 8 Output of the show aaa command for RADIUS

Field	Description
Radius key	The setting configured with the **radius-server key** command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Radius retries	The setting configured with the **radius-server retransmit** command.
Radius timeout	The setting configured with the **radius-server timeout** command.
Radius dead-time	The setting configured with the **radius-server dead-time** command.
Radius Server	For each RADIUS server, the IP address, and the following statistics are displayed: **Auth Port** – RADIUS authentication port number (default 1645) **Acct Port** – RADIUS accounting port number (default 1646) **opens** – Number of times the port was opened for communication with the server **closes** – Number of times the port was closed normally **timeouts** – Number of times port was closed due to a timeout **errors** – Number of times an error occurred while opening the port **packets in** – Number of packets received from the server **packets out** – Number of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.

The **show web** command displays the privilege level of Web Management Interface users.

```
NetIron(config)# show web
User                              Privilege       IP address
set                                      0       192.168.1.234
```

Syntax: show web

Configuring AAA on the console

Only enable-level authentication is available on the console by default. Command authorization and accounting and exec accounting must be explicitly configured. To enable AAA support on the console, use the following command.

```
NetIron(config)# enable aaa console
```

Syntax: [no] enable aaa console

After this command is added, use the following procedure to test the configuration.

1. At the console, type "**end**" to go to the Privileged EXEC level.
2. Type "**exit**" to go to the User EXEC level.

   After the enable **aaa console** command is, a new command, "**exit**", is be available at the User EXEC level.
3. Enter "**exit**" to display the following login prompt on the console window.

   ```
   "Press Enter key to login".
   ```
4. Press the "Enter key," to begin the login process.

   The next prompt to appear is determined by the first method configured in the login authentication configuration. If it is not TACACS+, the default prompts are used.

NOTE
If you use the use the **aaa console** command to enable AAA, you must make sure that the method lists are configured to allow access. Otherwise, you will be locked out of the console.

Configuring AAA authentication-method lists for login

With AAA is enabled on the console, you must configure an authentication-method list to set the conditions for granting access to the console. The authentication methods supported on the PowerConnect devices include the following:

- enable
- line
- local
- radius
- tacacs
- tacacs+
- none

When a list is configured, the first method listed is attempted to provide authentication at login. If that method is not available, (for example, a TACACs server can not be reached) the next method is tried until a method in the list is available or all methods have been tried. You can place the method none at the end of a list to ensure that access will always be available if all active methods fail.

To configure a AAA authentication-method list for login, use the following command.

```
NetIron(config)# aaa authentication login default tacacs+ local none
```

In this configuration, tacacs+ would be tried first. If a tacacs+ server cannot be reached, the local system password would be used. If this method fails, authentication would default to none.

Syntax: [no] aaa authentication login default enable line local none radius tacacs tacacs+

The **enable** option uses the enable password configured on the router to grant access to the console.

The **line** option uses the line password configured on the router to grant access to the console.

The **local** option uses the local password configured on the router to grant access to the console.

The **radius** option uses authentication provided by a radius server to grant access to the console.

The **tacacs** option uses authentication provided by a tacacs server to grant access to the console.

The **tacacs+** option uses authentication provided by a tacacs+ server to grant access to the console.

The **none** option eliminates the requirement for any authentication method to grant access to the console.

Configuring authentication-method lists

To implement one or more authentication methods for securing access to the device, you configure authentication-method lists that set the order in which the authentication methods are consulted.

In an authentication-method list, you specify the access method (Telnet, Web, SNMP, and so on) and the order in which the device tries one or more of the following authentication methods:

- Local Telnet login password
- Local password for the Super User privilege level
- Local user accounts configured on the device
- Database on a TACACS or TACACS+ server
- Database on a RADIUS server
- No authentication

NOTE
The TACACS or TACACS+, RADIUS, and Telnet login password authentication methods are not supported for SNMP access.

NOTE
To authenticate Telnet access to the CLI, you also must enable the authentication by entering the **enable telnet authentication** command at the global CONFIG level of the CLI. You cannot enable Telnet authentication using the Web Management Interface.

NOTE
You do not need an authentication-method list to secure access based on ACLs or a list of IP addresses. Refer to "Using ACLs to restrict remote access" on page 20 or "Restricting remote access to the device to specific IP addresses" on page 23.

In an authentication-method list for a particular access method, you can specify up to seven authentication methods. If the first authentication method is successful, the software grants access and stops the authentication process. If the access is rejected by the first authentication method, the software denies access and stops checking.

However, if an error occurs with an authentication method, the software tries the next method on the list, and so on. For example, if the first authentication method is the RADIUS server, but the link to the server is down, the software will try the next authentication method in the list.

NOTE
If an authentication method is working properly and the password (and user name, if applicable) is not known to that method, this is not an error. The authentication attempt stops, and the user is denied access.

The software will continue this process until either the authentication method is passed or the software reaches the end of the method list. If the Super User level password is not rejected after all the access methods in the list have been tried, access is granted.

NOTE
If a user cannot be authenticated using local authentication, then the next method on the authentication methods list is used to try to authenticate the user. If there is no method following local authentication, then the user is denied access to the device.

Configuration considerations for authentication-method lists

The configuration considerations for authentication-method lists are as follows:

- For CLI access, you must configure authentication-method lists if you want the device to authenticate access using local user accounts or a RADIUS server. Otherwise, the device will authenticate using only the locally based password for the Super User privilege level.
- When no authentication-method list is configured specifically for Web management access, the device performs authentication using the SNMP community strings:
  - For read-only access, you can use the user name “get” and the password “public”. The default read-only community string is “public”.
  - There is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web Management Interface. You first must configure a read-write community string using the CLI. Then you can log on using “set” as the user name and the read-write community string you configure as the password. Refer to “Configuring TACACS or TACACS+ security” on page 39.
- If you configure an authentication-method list for Web management access and specify “local” as the primary authentication method, users who attempt to access the device using the Web Management Interface must supply a user name and password configured in one of the local user accounts on the device. The user ***cannot*** access the device by entering “set” or “get” and the corresponding SNMP community string.

- For devices that can be managed using SNMP, the default authentication method (if no authentication-method list is configured for SNMP) is the CLI Super User level password. If no Super User level password is configured, then access through IronView Network Manager is not authenticated. To use local user accounts to authenticate access through SNMP, configure an authentication-method list for SNMP access and specify “local” as the primary authentication method.

Examples of authentication-method lists

The following example shows how to configure authentication-method lists for the Web Management Interface, SNMP, and the Privileged EXEC and CONFIG levels of the CLI. In this example, the primary authentication method for each is “local”. The device will authenticate access attempts using the locally configured user names and passwords first.

To configure an authentication-method list for the Web Management Interface, enter a command such as the following.

```
NetIron(config)# aaa authentication web-server default local
```

This command configures the device to use the local user accounts to authenticate access to the device through the Web Management Interface. If the device does not have a user account that matches the user name and password entered by the user, the user is not granted access.

To configure an authentication-method list for SNMP Network Manager, enter a command such as the following.

```
NetIron(config)# aaa authentication snmp-server default local
```

This command configures the device to use the local user accounts to authenticate access attempts through any network management software, such as SNMP Network Manager.

To configure an authentication-method list for the Privileged EXEC and CONFIG levels of the CLI, enter the following command.

```
NetIron(config)# aaa authentication enable default local
```

This command configures the device to use the local user accounts to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI.

Example

To configure the device to consult a RADIUS server first to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI, then consult the local user accounts if the RADIUS server is unavailable, enter the following command.

```
NetIron(config)# aaa authentication enable default radius local
```

Syntax: [no] aaa authentication snmp-server | web-server | enable | login | dot1x default <*method1*> [<*method2*>] [<*method3*>] [<*method4*>] [<*method5*>] [<*method6*>] [<*method7*>]

The **snmp-server | web-server | enable | login | dot1x** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE
If you configure authentication for Web management access, authentication is performed each time a page is requested from the server. When frames are enabled on the Web Management Interface, the browser sends an HTTP request for each frame. The Dell device authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web Management Interface.

NOTE
TACACS or TACACS+ and RADIUS are not supported with the **snmp-server** parameter.

The <*method1*> parameter specifies the primary authentication method. The remaining optional <*method*> parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in Table 9.

TABLE 9 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the **enable telnet password...** command. Refer to "Setting a Telnet password" on page 27.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password...** command. Refer to "Setting passwords for management privilege levels" on page 27.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the **username...** command. Refer to "Configuring a local user account" on page 31.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the **tacacs-server** command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the **tacacs-server** command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the **radius-server** command.
none	Do not use any authentication method. The device automatically permits access.

Configuring Basic Parameters 3

Overview

The following Basic Parameters features are supported by NetIron MLX Series devices:

- Simple Network Management (SNMP) traps
- SNMP ifIndex
- Optical Monitoring
- Optics Compatibility Checking
- New encryption code for passwords, authentication keys, and community strings
- Designating an Interface as the source for packets
- Simple Network Time Protocol (SNTP) server
- Setting the systemclock
- Command Alias
- Limiting broadcast, multicast, or unknown-unicast rates
- CLI banners
- Terminal display
- Modifying system parameter default settings
- System low memory prevention and reporting
- Layer 2 switching
- MAC age time
- Static ARP entries
- Configurable CAM size for IPv4 and IPv6 multicast entries
- Switch fabric fault monitoring
- Automatic switch fabric module shutdown
- Switch fabric utilization monitoring
- Verifying an image checksum
- Displaying information for an interface for an Ethernet port
- Displaying statistics Information for an Ethernet port
- Real-time monitoring

This chapter describes how to configure basic system parameters.

NOTE
For information about the Syslog buffer and messages, refer to Appendix A, “Using Syslog”.

The PowerConnect router is configured with default parameters to allow you to begin using the basic features of the system immediately. However, many advanced features, such as VLANs or routing protocols for the router, must first be enabled at the system (global) level before they can be configured.

You can find system level parameters at the Global CONFIG level of the CLI.

NOTE
Before assigning or modifying any router parameters, you must assign the IP subnet (interface) addresses for each port.

Entering system administration information

You can configure a system name, contact, and location for the PowerConnect router and save the information locally in the configuration file for future reference. The information is not required for system operation but recommended. When you configure a system name, it replaces the default system name in the CLI command prompt.

To configure a system name, contact, and location, enter commands such as the following.

```
NetIron(config)# hostname home
home(config)# snmp-server contact Suzy Sanchez
home(config)# snmp-server location Centerville
home(config)# end
home# write memory
```

The system name you configure **home** replaces the system name PowerConnect.

Syntax: [no] hostname <*string*>

Syntax: [no] snmp-server contact <*string*>

Syntax: [no] snmp-server location <*string*>

The name, contact, and location each can be up to 255 alphanumeric characters. The text strings can contain blanks. The SNMP text strings do not require quotation marks when they contain blanks but the host name does.

Configuring Simple Network Management (SNMP) traps

This section explains how to do the following:

- Specify an SNMP trap receiver.
- Specify a source address and community string for all traps that the PowerConnect router sends.
- Change the holddown time for SNMP traps.
- Disable individual SNMP traps. (All traps are enabled by default.)
- Disable traps for CLI access that is authenticated by a local user account, a RADIUS server, or a TACACS or TACACS+ server.

NOTE
To add and modify "get" (read-only) and "set" (read-write) community strings, refer to "Securing Access to Management Functions" on page 17.

Specifying an SNMP trap receiver

You can specify a trap receiver to ensure that all SNMP traps sent by the PowerConnect router go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. When you specify the host, you also specify a community string. The PowerConnect router sends all the SNMP traps to the specified hosts and includes the specified community string. Administrators can therefore filter for traps from a PowerConnect router based on IP address or community string.

If the string is in the clear format, the system will internally encrypt it. When you display or save the configuration, the encrypted string is used.

To specify an SNMP trap receiver, enter a command such as the following.

```
NetIron(config)# snmp-server host 2.2.2.2 mypublic port 200
```

The command adds trap receiver 2.2.2.2 and designates the UDP port that will be used to receive traps.

Syntax: [no] snmp-server host <*ip-addr*> <*string*> **[port** <*value*>**]**

The <*ip-addr*> parameter specifies the IP address of the trap receiver.

The <*string*> parameter specifies an SNMP community string configured on the PowerConnect router. It is not used to authenticate access to the trap host, but it is a useful method for filtering traps on the host. For example, if you configure each of your PowerConnect devices that use the trap host to send a different community string, you can easily distinguish among the traps from the devices based on the community strings.

By default, <*string*> is encrypted. If you want <*string*> to be in clear text, insert a **0** preceding <*string*>.

Example

```
NetIron(config)# snmp-server host 2.2.2.2 0 mypublic port 200
```

The software adds a prefix to the string in the configuration. For example, the following portion of the code has the encrypted code "2".

```
snmp-server host 2.2.2.2 version v2c 2 $Si2^=d
```

The prefix can be one of the following:

- 0 = the key string is not encrypted and is in clear text
- 1 = the key string uses simple encryption
- 2 = the key string uses base64 encryption format

The **port** <*value*> parameter specifies the UDP port that will be used to receive traps. This parameter allows you to configure several trap receivers in a system. With this parameter, SNMP Network Manager and another network management application can coexist in the same system. The PowerConnect devices can be configured to send copies of traps to more than one network management application.

Specifying a single trap source

You can specify a single trap source to ensure that all SNMP traps sent by the PowerConnect router use the same source IP address. When you configure the SNMP source address, you specify the Ethernet port, loopback interface, or virtual routing interface that is the source for the traps. The PowerConnect router then uses the lowest-numbered IP address configured on the port or interface as the source IP address in the SNMP traps it sends.

Identifying a single source IP address for SNMP traps provides the following benefits:

- If your trap receiver is configured to accept traps only from specific links or IP addresses, you can simplify configuration of the trap receiver by configuring the PowerConnect router to always send the traps from the same link or source address.
- If you specify a loopback interface as the single source for SNMP traps, SNMP trap receivers can receive traps regardless of the states of individual links. Thus, if a link to the trap receiver becomes unavailable but the receiver can be reached through another link, the receiver still receives the trap, and the trap still has the source IP address of the loopback interface.

To configure the PowerConnect router to send all SNMP traps from the first configured IP address on port 4/11, enter the following commands.

```
NetIron(config)# snmp-server trap-source ethernet 4/11
NetIron(config)# write memory
```

Syntax: [no] snmp-server trap-source loopback <*num*> **| ethernet** <*slot/port*> **| ve** <*num*>

The <*num*> parameter is a loopback interface or virtual routing interface number.

If you do not configure this command, the device will use the device router ID as the source IP address of the notification packet. The router ID of the device can be obtained from the “show ip” command output.

To specify a loopback interface as the device’s SNMP trap source, enter following commands.

```
NetIron(config)# int loopback 1
NetIron(config-lbif-1)# ip address 10.0.0.1/24
NetIron(config-lbif-1)# exit
NetIron(config)# snmp-server trap-source loopback 1
```

The commands configure loopback interface 1, gives it IP address 10.00.1/24, then designate it as the SNMP trap source for the PowerConnect router. Regardless of the port the PowerConnect uses to send traps to the receiver, the traps always arrive from the same source IP address.

Setting the SNMP trap holddown time

When a PowerConnect router starts up, the software waits for Layer 2 convergence (STP) and Layer 3 convergence (OSPF) before beginning to send SNMP traps to external SNMP servers. Until convergence occurs, the PowerConnect router might not be able to reach the servers, in which case the messages are lost.

By default, the PowerConnect router uses a one-minute holddown time to wait for the convergence to occur before starting to send SNMP traps. After the holddown time expires, the PowerConnect router sends the traps, including traps such as “cold start” or “warm start” that occur before the holddown time expires.

You can change the holddown time to a value from one second to ten minutes.

To change the holddown time for SNMP traps, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# snmp-server enable traps holddown-time 30
```

The command changes the holddown time for SNMP traps to 30 seconds. The PowerConnect router waits 30 seconds to allow convergence in STP and OSPF before sending traps to the SNMP trap receiver.

Syntax: [no] snmp-server enable traps holddown-time <*secs*>

The <*secs*> parameter specifies the number of seconds (1 – 600). The default is 60.

Disabling SNMP traps

The PowerConnect router comes with SNMP trap generation enabled by default for all traps.

NOTE
By default, all SNMP traps are enabled at system startup.

You can selectively disable one or more of the following traps:

- SNMP authentication key
- Temperature
- Power supply failure
- Fan failure
- Cold start
- Link up
- Link down
- Bridge new root
- Bridge topology change
- Module insert
- Module remove
- Redundant module
- Metro-ring
- MPLS
- BGP4
- OSPF
- VRRP
- VSRP

To stop link down occurrences from being reported, enter the following command.

```
NetIron(config)# no snmp-server enable traps link-down
```

Syntax: [no] snmp-server enable traps <*trap-type*>

A list of traps is available in the *IronWare MIB Reference*.

Disabling Syslog messages and traps for CLI access

The PowerConnect router sends Syslog messages and SNMP traps when a user logs into or out of the User EXEC or Privileged EXEC level of the CLI. The feature, enabled by default, applies to users whose access is determined by the AAA engine.

NOTE
The Privileged EXEC level is sometimes called the "Enable" level, because the command for accessing this level is **enable**.

Examples of Syslog messages for CLI access

When a user whose access is authenticated by a local user account, a RADIUS server, or a TACACS or TACACS+ server logs into or out of the CLI's User EXEC or Privileged EXEC mode, the software generates a Syslog message and trap containing the following information:

- The time stamp
- The user name
- Whether the user logged in or out
- The CLI level the user logged into or out of (User EXEC or Privileged EXEC level)

NOTE
Messages for accessing the User EXEC level apply only to access through Telnet. The device does not authenticate initial access through serial connections but does authenticate serial access to the Privileged EXEC level. Messages for accessing the Privileged EXEC level apply to access through the serial connection or Telnet.

The following examples show login and logout messages for the User EXEC and Privileged EXEC levels of the CLI.

```
NetIron(config)# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
Buffer logging: level ACDMEINW, 12 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning
Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Oct 15 18:01:11:info:dg logout from USER EXEC mode
Oct 15 17:59:22:info:dg logout from PRIVILEGE EXEC mode
Oct 15 17:38:07:info:dg login to PRIVILEGE EXEC mode
Oct 15 17:38:03:info:dg login to USER EXEC mode
```

Syntax: show logging

The first message (the one on the bottom) indicates that user "dg" logged in to the CLI's User EXEC level on October 15 at 5:38 PM and 3 seconds (Oct 15 17:38:03). The same user logged into the Privileged EXEC level four seconds later.

The user remained in the Privileged EXEC mode until 5:59 PM and 22 seconds. (The user could have used the CONFIG modes as well. Once you access the Privileged EXEC level, no further authentication is required to access the CONFIG levels.) At 6:01 PM and 11 seconds, the user ended the CLI session.

Example : Direct Telnet access without a password check.

```
00:48:53:I:Security: telnet login from src IP 172.16.2.100 to USER EXEC mode
00:48:59:I:Security: telnet logout from src IP 172.16.2.100 to USER EXEC mode
```

Example : Telnet access using a local user.

```
Oct 30 01:06:30:I:Security: telnet login by sunil from src IP 172.16.2.100 to
USER EXEC mode
```

Example : SSH access.

```
Oct 30 01:08:58:I:Security: ssh login by sunil from src IP 172.16.2.100 to USER
EXEC mode
```

Disabling the Syslog messages and traps

Logging of CLI access is enabled by default. To disable logging of CLI access, enter the following commands.

```
NetIron(config)# no logging enable user-login
```

Syntax: [no] logging enable user-login

Refer to the MIB Guide for a list of traps.

Configuring SNMP ifIndex

This section explains how ifIndex values are assigned on PowerConnect devices.

On NetIron MLX Series devices only

On **NetIron MLX Series** devices, SNMP Management Information Base (MIB) uses Interface Index (ifIndex) to assign a unique value to each port on a module or slot. The number of indexes that can be assigned per module is 20, 40, or 64, depending on the number of ports on the module.

Enter the following to change the number of indexes per module.

```
NetIron(config)# snmp-server max-ifindex-per-module 40
```

Syntax: [no] snmp-server max-ifindex-per-module [20 | 40 | 64]

20 is the default.

You cannot change the maximum ifIndex per module to a number less than the number of ports.

After this command is issued the following are generated:

- “System: IfIndex assignment was changed” is logged in the Syslog.
- The snTrapIfIndexAssignmentChanged trap is generated.

Configuration notes for the Product Name

Note the following if you are upgrading the software on the Product Name:

- If you are running an earlier version of the software and you will not be installing the NI-MLX-1Gx48-T module, you do not need to change your ifIndex allocation scheme. The current definition is maintained. The maximum ifIndex per module can remain at 20 or 40.
- If you are running an earlier version of the software and you will be installing the NI-MLX-1Gx48-T module on you Product Name, you must configure the maximum ifIndex per module to 64. **You must change the ifIndex allocation before installing the NI-MLX-1Gx48-T module**; otherwise, the module status remains in the Offline state.
- If you have a new Product Name (no previous software installed), but will not be installing an NI-MLX-1Gx48-T module, it is recommended that you configure the maximum ifIndex per module to 64 to avoid future ifIndex problems in case an NI-MLX-1Gx48-T module is installed in the future.
- If you have a new Product Name (no previous software installed), and you will be installing an NI-MLX-1Gx48-T module, you **must** configure the maximum ifIndex per module to 64; otherwise, the module remains in the Offline state.

Configuring optical monitoring

You can configure your PowerConnect router to monitor XFPs or SFPs in the system either globally or by specified port. If monitoring is enabled, console messages, syslog messages, and SNMP traps are sent when XFP or SFP operating conditions warrant it and a port is enabled.

Configure all XFP and SFP ports for optical monitoring, using the following command.

```
NetIron(config)# optical-monitor
```

Configure a specific XFP or SFP port for optical monitoring, using the following command.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# optical-monitor
```

Configure a range of XFP or SFP ports for optical monitoring, using the following command.

```
NetIron(config)# interface ethernet 1/1 to 1/2
NetIron(config-mif-e10000-1/1-1/2)# optical-monitor
```

Syntax: [no] optical-monitor <*alarm-interval*>

The optional <*alarm-interval*> variable sets the interval in minutes between which alarms or messages are sent. The default interval is 3 minutes.

You can view the XFP optical monitoring information using the show optic command as displayed in the following.

```
NetIron#show optic 4
 Port  Temperature   Tx Power     Rx Power     Tx Bias Current
+----+-----------+----------+------------+------------------+
4/1   30.8242 C  -001.8822 dBm -002.5908 dBm   41.790 mA
       Normal       Normal         Normal         Normal
4/2   31.7070 C  -001.4116 dBm -006.4092 dBm   41.976 mA
       Normal       Normal         Normal         Normal
4/3   30.1835 C                -000.5794 dBm    0.000 mA
       Normal      Low-Alarm      Normal        Low-Alarm
4/4    0.0000 C                                 0.000 mA
       Normal       Normal         Normal         Normal
```

For Temperature, Tx Power, Rx Power, and Tx Bias Current, values are displayed along with one of the following status values: Low-Alarm, Low-Warn, Normal, High-Warn or High-Alarm. The thresholds that determine these status values are set by the manufacturer of the XFPs. Table 10 describes each of these status values.

TABLE 10 Status value description

Status value	Description
Low-alarm	The monitored level has dropped below the "low-alarm" threshold set by the XFP or SFP manufacturer.
Low-warn	The monitored level has dropped below the "low-warn" threshold set by the XFP or SFP manufacturer.
Normal	The monitored level is within the "normal" range" set by the XFP or SFP manufacturer.
High-warn	The monitored level has climbed above the "high-warn" threshold set by the XFP or SFP manufacturer.
High-alarm	The monitored level has climbed above the "high-alarm" threshold set by the XFP or SFP manufacturer.

NOTE
This function takes advantage of information stored and supplied by the SFP or XFP device. This information is an optional feature of the Multi-Source Agreement standard defining the SFP or XFP interface. Not all component suppliers have implemented this feature set. In such cases where the SFP or XFP device does not supply the information, a “Not Available” message will be displayed for the specific port that the device is installed

Displaying media information

To display media information for SFP and XFP devices installed in a specific slot, enter the following command at any CLI level.

```
NetIron#show media slot 3
Port 2/1:
    Type  : 10GBASE-SR/SW 850.00nm (XFP)
    Vendor:     FOUNDRY NETWORKS, Version:                    00
    Part# :     FTLX8511D3-F1   , Serial#:     A9C01ED
Port 2/2:
    Type  : 10GBASE-SR 850.00nm (XFP)
    Vendor:     Foundry Networks, Version:                    01
    Part# :     AFBR-720XPDZ-FD1, Serial#:     AVAGCN0933MD10U
.
.
```

```
.
All show media done
```

The example above displays all optical devices on slot 3.

Syntax: show media slot <*slot-number*>

To display media information for SFP and XFP devices installed in an ethernet port, enter the following command at any CLI level.

```
NetIron#show media ethernet 3/4
Port 3/4:
    Type  : 10GBASE-SR/SW 854.00nm (XFP)
    Vendor:     Foundry Networks, Version:                   02
    Part# :     JXPR01SW05306   , Serial#:     F74340380372
```

Syntax: show media [ethernet <*slot-port*> **[to** <*slot-port*>**]]**

You can display media information for all ports in an NetIron router by using the **show media** command without options.

The **ethernet** <*slot-port*> parameter limits the display to a single port.

The **to** <*slot-port*> parameter displays information for a range of ports.

This results displayed from this command provide the Type, Vendor, Part number, Version and Serial number of the SFP or XFP optical device installed in the port.

If no SFP or XFP device is installed in a port, the “Type” field will display “N/A”, the “Vendor” field will be empty and the other fields will display “Unknown”.

Multi-rate optical transceivers are supported. In this case, if a multi-rate optical transceiver is inserted in an Interface module, the “Type” parameter will display the transmission code for the correct value for the port as determined by either the Interface module type or the configuration of the port. There is one exception to this rule however. If a port is in the disabled state only one type will be displayed. Once the port is enabled, the correct “Type” will be displayed in accordance with the configuration.

Optics compatibility checking

This feature checks the installation of the following optical transceivers into Interface module ports and shuts down the port if the transceiver is incompatible with the port:

- **10 GbE XFP** – This interface is brought up if the XFP is compliant with Ethernet transmission compliance.
- **10 Gbe SFP+** - This interface is brought up if the SFP+ is compliant with Ethernet transmission.
- 1 Gb (100/1000) Ethernet interface will be enabled if the SFP is Ethernet capable.

If the interface is incompatible with the optical transceiver installed, the port will not come up and the syslog message “**Incompatible optical trans-receiver detected on port** <*n*>” is displayed. An SNMP trap is also generated and the port is described as “down” because of “(incompatible transceiver)” in the output from the **show interface** command.

Multi-rate optical transceivers (XFP and SFP) are supported as described in the following:

- In Multi-rate SFPs and XFPs, the EEPROM is programmed for multi-rate.
- Multi-rate SFPs and XFPs are supported. The system software checks for transmission compatibility against the interface configuration.

- The **show media** command described in "Configuring optical monitoring" on page 84 continues to show only one transmission rate even for multi-rate SFPs and XFPs. If the interface is enabled and the SFP or XFP is compatible, the **show media** command only displays the compatible transmission code in the "Type" field. If the interface is disabled, the **show media** display depends on the module type. For Ethernet interface modules, the Ethernet compliance code is shown.

Disabling transceiver type checking

When transceiver type checking is disabled, the syslog message "**Incompatible optical trans-receiver detected on port** <*n*>" is still displayed but the port is not shut down. You can disable transceiver type checking with the **no transceiver-type-check** command as shown in the following.

```
NetIron(config)# no transceiver-type-check
```

Syntax: [no] transceiver-type-check

Transceiver type checking is on by default and the command is not included in the configuration.

The **no** option of the **transceiver-type-check** command, disables transceiver type checking as described, sends a syslog message and places the command in the configuration.

Using the **transceiver-type-check** command without the no option, enables transceiver type checking, sends a syslog message and removes the command from the configuration.

Designating an interface as the packet source

The software uses the lowest-numbered IP address configured on an interface as the source IP address for all Telnet, SSH, SNTP, TFTP, TACACS or TACACS+, or RADIUS packets originated from the PowerConnect router.

You can specify the source interface for one or more of these types of packets.

Configuring an interface as the source for all Telnet packets

Identifying a single source IP address for Telnet packets provides the following benefits:

- If your Telnet server is configured to accept packets only from specific links or IP addresses, you can simplify configuration of the Telnet server by configuring the PowerConnect router to always send the Telnet packets from the same link or source address.
- If you specify a loopback interface as the single source for Telnet packets, Telnet servers can receive the packets regardless of the states of individual links. Thus, if a link to the Telnet server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

To specify the lowest-numbered IP address configured on a loopback interface as the device's source for all Telnet packets, enter commands such as the following.

```
NetIron(config)# int loopback 2
NetIron(config-lbif-2)# ip address 10.0.0.2/24
NetIron(config-lbif-2)# exit
NetIron(config)# ip telnet source-interface loopback 2
```

The commands configure loopback interface 2, assign IP address 10.0.0.2/24 to it, then designate it as the source for all Telnet packets from the PowerConnect router.

Syntax: [no] ip telnet source-interface ethernet <*portnum*> **| loopback** <*num*> **| ve** <*num*>

The following commands configure an IP interface on an Ethernet port and designate the address port as the source for all Telnet packets from the PowerConnect router.

```
NetIron(config)# interface ethernet 1/4
NetIron(config-if-e10000-1/4)# ip address 209.157.22.110/24
NetIron(config-if-e10000-1/4)# exit
NetIron(config)# ip telnet source-interface ethernet 1/4
```

Cancelling an outbound Telnet session

If you want to cancel a Telnet session from the console to a remote Telnet server (for example, if the connection is frozen), you can terminate the Telnet session by performing the tasks listed below.

1. At the console, press Ctrl-^ (Ctrl-Shift-6).
2. Press the X key to terminate the Telnet session.

Pressing Ctrl-^ twice in a row causes a single Ctrl-^ character to be sent to the Telnet server. After you press Ctrl-^, pressing any key other than X or Ctrl-^ returns you to the Telnet session.

Configuring an interface as the source for all SSH packets

You can configure the PowerConnect router to use the lowest-numbered IP address configured on a loopback interface, virtual routing interface, or Ethernet port as the source for the SSH packets it sends.

For example, to specify an Ethernet port as the interface whose lowest-numbered IP address will be the source address for the SSH packets originated from the PowerConnect router, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/5
NetIron(config-if-e10000-1/5)# ip address 209.157.22.111/24
NetIron(config-if-e10000-1/5)# exit
NetIron(config)# ip ssh source-interface ethernet 1/5
```

The commands configure Ethernet port 1/5, assign IP address 209.157.22.111/24 to it, then designate it as the source interface.

Syntax: [no] ip ssh source-interface ethernet <*portnum*> **| loopback** <*num*> **| ve** <*num*>

Configuring an interface as the source for all SNTP packets

You can configure the PowerConnect router to use the lowest-numbered IP address configured on a loopback interface, virtual routing interface, or Ethernet port as the source for the SNTP packets it sends.

For example, to specify a virtual routing interface as the interface whose lowest-numbered IP address will be the source address for the SNTP packets originated from the PowerConnect router, enter commands such as the following.

```
NetIron(config)# int ve 1
NetIron(config-vif-1)# ip address 10.0.0.3/24
NetIron(config-vif-1)# exit
NetIron(config)# ip sntp source-interface ve 1
```

The commands configure virtual routing interface 1, assign IP address 10.0.0.3/24 to it, then designate it as the source interface.

Syntax: [no] ip sntp source-interface ethernet <*portnum*> **| loopback** <*num*> **| ve** <*num*>

Configuring an interface as the source for all TFTP packets

You can configure the PowerConnect router to use the lowest-numbered IP address configured on a loopback interface, virtual routing interface, or Ethernet port as the source for all TFTP packets it sends.

For example, to specify the lowest-numbered IP address configured on a virtual routing interface as the PowerConnect's source for all TFTP packets, enter commands such as the following.

```
NetIron(config)# int ve 1
NetIron(config-vif-1)# ip address 10.0.0.3/24
NetIron(config-vif-1)# exit
NetIron(config)# ip tftp source-interface ve 1
```

The commands configure virtual routing interface 1, assign IP address 10.0.0.3/24 to it, then designate the address as the source address for all TFTP packets.

Syntax: [no] ip tftp source-interface ethernet <*portnum*> **| loopback** <*num*> **| ve** <*num*>

The default is the lowest-numbered IP address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

Configuring an interface as the source for all TACACS or TACACS+ packets

You can configure the PowerConnect router to use the lowest-numbered IP address configured on a loopback interface, virtual routing interface, or Ethernet port as the source for the TACACS or TACACS+ packets it sends.

For example, to specify a virtual routing interface as the interface whose lowest-numbered IP address will be the source address for the TACACS or TACACS+ packets originated from the PowerConnect router, enter commands such as the following.

```
NetIron(config)# int ve 1
NetIron(config-vif-1)# ip address 10.0.0.3/24
NetIron(config-vif-1)# exit
NetIron(config)# ip tacacs source-interface ve 1
```

The commands configure virtual routing interface 1, assign IP address 10.0.0.3/24 to it, then designate it as the source interface.

Syntax: [no] ip tacacs source-interface ethernet <*portnum*> **| loopback** <*num*> **| ve** <*num*>

Configuring an interface as the source for all RADIUS packets

You can configure the PowerConnect router to use the lowest-numbered IP address configured on a loopback interface, virtual routing interface, or Ethernet port as the source for the RADIUS packets it sends.

For example, to specify an Ethernet port as the interface whose lowest-numbered IP address will be the source address for the RADIUS packets originated from the PowerConnect router, enter the following commands.

```
NetIron(config)# interface ethernet 1/5
NetIron(config-if-e10000-1/5)# ip address 209.157.22.111/24
NetIron(config-if-e10000-1/5)# exit
NetIron(config)# ip radius source-interface ethernet 1/5
```

The commands configure Ethernet port 1/5, assign IP address 209.157.22.111/24 to it, then designate it as the source interface.

Syntax: **[no] ip radius source-interface ethernet** <*portnum*> **| loopback** <*num*> **| ve** <*num*>

Specifying a Simple Network Time Protocol (SNTP) server

The PowerConnect can be configured both as an SNTP server and as an SNTP client. This section describes how to configure the PowerConnect device as an SNTP client, consulting upstream SNTP servers for the current system time and date.

You can configure the device to consult up to 3 SNTP servers to establish the current system time and date. The first server configured will be used unless it becomes unreachable, in which case the device will attempt to synchronize with the other SNTP servers (if any) in the order in which they were configured.

NOTE
While the PowerConnect device does retain time and date information across power cycles, the system time can drift and should ideally be synchronized with a reliable external clock source. Therefore Dell recommends that you use the SNTP feature as described below.

To identify an SNTP server with IP address 208.99.8.95 to act as the clock reference for a PowerConnect, router enter the following command.

```
NetIron(config)# sntp server 208.99.8.95
```

Syntax: **[no] sntp server** { <*ip-address*> | <*hostname*> | **ipv6** <*ipv6-address*> } [<*sntp-version*>] [**authentication-key** <*key-ID*> <*key-string*>]

- The <*sntp-version*> parameter specifies the SNTP version the server is running and can be from 1 – 4. The default is 4. You can configure up to three SNTP servers by entering three separate **sntp server** commands.
- The order in which the SNTP servers are configured is the order in which they are consulted. The server that was configured first is the first server consulted after the poll cycle; the next server will be consulted only if a positive ACK is not received from the first one.

- To specify an IPv6 address for the SNTP server, use the **ipv6** option.
- The **authentication-key** option allows you to configure an authentication key for communication with the SNTP server. The <*key-ID*> is the symmetric key shared with the upstream server, and accepts values from 1 to 4,294,967,295. The <*key-string*> is the authentication string itself, and can take up to 16 characters.

Modification of the authentication key fields is not supported. To change the key ID or key string, remove the time server using the **no sntp server...** command, then reconfigure the server with the new key.

```
NetIron(config)# sntp poll-interval 900
```

Syntax: [no] sntp poll-interval <*1-65535*>

To display information about SNTP associations, enter the following command.

```
NetIron# show sntp associations
  address                 ref clock       st  when  poll    delay     disp
*~10.50.2.121             10.60.2.100      3     4  1800    0.000    0.000
 ~216.218.254.202         CDMA             1   119  1800    0.000    0.000
 ~10.51.3.123             0.0.0.0         16    -1  1800    0.000    0.000
* synced, ~ configured
```

Syntax: show sntp associations

The following table describes the information displayed by the **show sntp associations** command.

TABLE 11 Output from the show sntp associations command

This field...	Displays...
(leading character)	One or both of the following: * Synchronized to this peer ~ Peer is statically configured
address	IP address of the peer
ref clock	IP address of the peer's reference clock, or the reference ID of the external clock source if the peer is stratum 1. Examples of external clock source IDs: GPS, CDMA, WWV (Ft.Collins US Radio 2.5, 5, 10, 15 MHz), CESM (calibrated Cesium clock), etc.
st	NTP stratum level of the peer
when	Amount of time since the last NTP packet was received from the peer. A negative number indicates the system has never received any synchcronization message from the specified server.
poll	Poll interval in seconds
delay	Round trip delay in milliseconds
disp	Dispersion in seconds

To display information about SNTP status, enter the following command.

```
NetIron# show sntp status
Clock is synchronized, stratum = 3, reference clock = 10.50.2.121
precision is 2**-20
reference time is 3478372757.0
clock offset is 144.23966407 msec, root delay is 0.000 msec
root dispersion is 0.000 msec, peer dispersion is 0.000 msec
```

Syntax: show sntp status

The following table describes the information displayed by the **show sntp status** command.

TABLE 12 Output from the show sntp status command

This field...	Indicates...
unsynchronized	System is not synchronized to an NTP peer.
synchronized	System is synchronized to an NTP peer.
stratum	NTP stratum level of the upstream time server
reference clock	IP Address of the peer (if any) to which the unit is synchronized
precision	Precision of this system's clock (in Hz)
reference time	Reference time stamp
clock offset	Offset of clock to synchronized peer
root delay	Total delay along the path to the root clock
root dispersion	Dispersion of the root path
peer dispersion	Dispersion of the synchronized peer

Configuring the device as an SNTP server

You can configure the PowerConnect device to function as an SNTP server to its downstream clients. When using the device as an SNTP server, you can also set it to use its own internal clock as the reference source if an upstream server becomes unavailable.

To use the device as a an SNTP server, enter a command such as the following at the Privileged EXEC level.

```
NetIron(config)# sntp server-mode use-local-clock authentication-key abc123
NetIron(config)# write memory
```

The above example configures the device to operate as an SNTP server with the local clock as a reference backup and an authentication key of “abc123” and writes the configuration changes to memory.

Syntax: [**no**] **sntp server-mode** [**use-local-clock** [**stratum** <*stratum-number*>]] [**authentication-key** <*key-string*>]

- The **use-local-clock** option causes the PowerConnect device to use the local clock as a reference source if an upstream reference source becomes unavailable. The SNTP stratum number is set to 1 by default. You may specify a different stratum number using the **stratum** option; <*stratum-number*> must be between 1 and 15. When the internal clock is serving as the SNTP reference source, the device will use the specified stratum number (or the default value of 1). When it is synchronized with the upstream server, the device will use the upstream server’s stratum number plus 1.
  If you do not include the **use-local-clock** option the device will function as specified by RFC 4330: when the device loses upstream synchronization, it will respond to client SNTP requests with a “kiss-of-death” response (stratum value=0).

NOTE
To enable the **use-local-clock** option, you must set the device internal clock either by SNTP synchronization (see "Specifying a Simple Network Time Protocol (SNTP) server" on page 90) or by using the **clock set** command (see "Setting the system clock" on page 94). Until the internal clock is set, the device will continue to rely exclusively on an upstream SNTP server if one is reachable. If none, the PowerConnect SNTP server is disabled (down).

- To require a code string for authentication of SNTP communication from clients, use the **authentication-key** option and enter a key string of up to 16 characters. When this option is used, authentication parameters are required in clients' SNTP request messages. If authentication fails, the device will reply with stratum 0 and a reference ID code of "CRYP" (cryptographic authentication or identification failed), and messages received without the required parameters will be dropped.

NOTE
Once entered, the authentication key cannot be viewed. Using the **show running-config** command will show output similar to the following when an authentication key has been set:

```
sntp server-mode authentication-key 2 $QHMiR3NzQA=
```

The **2** indicates that the key is encrypted using base-64 encryption; the characters following the 2 are the encrypted authentication string.

NOTE
You cannot enable or disable the **use-local-clock** option (or its stratum number) or change the authentication string when the SNTP server is up. To change these settings after enabling SNTP server mode, you must disable server mode using the command **no sntp server-mode**, then re-enable it with the new parameters.

Displaying SNTP server information

Use the following command to display the status of the SNTP server and its configuration.

```
NetIron# show sntp server-mode
Status            : up
Stratum           : 4
Authentication    : md5
Clock source      : 10.50.2.121
Last upstream sync: 15:55:00 Pacific Sun Jul 5 2009

Last 5 unique responses sent to downstream clients :
Client Address           Reference Time
10.1.50.23               16:10:32 Pacific Sun Jul 5 2009
10.1.52.34               15:50:40 Pacific Sun Jul 5 2009
10.1.50.41               10:22:08 Pacific Fri Jul 3 2009
10.1.50.10               06:21:03 Pacific Fri Jul 3 2009
10.1.50.29               21:17:39 Pacific Fri Jul 2 2009
```

Syntax: show sntp server-mode

TABLE 13 Output from the **show sntp server-mode** command

This field...	Displays...
status	The operational state of the SNTP server. "**Up**" means that the SNTP port is open; "**down**" means that the SNTP port is closed. (If sntp server-mode is disabled, the **show sntp server-mode** command will display the message "SNTP server is not operational.)
stratum	Stratum number of this server. If the PowerConnect is synchronized to an upstream SNTP server, this will show that server's stratum number +1. If the PowerConnect is unsynchronized and using the **use-local-clock** option, this will show the user-specified stratum number (or the default value of "**1**" if no stratum has been configured).
authentication	Authentication key used. If authentication has been configured successfully, this displays "**md5**." If not, it displays "**none**."
clock source	The source of the reference time. When the reference source is an upstream SNTP server, this will show the IP address of the upstream server. When the device internal clock is being used as the reference, this will show "**local-clock**."
last upstream sync	The last upstream time-server synchronization, displayed in timestamp format. This field is not displayed if the time source is the local clock.
last responses sent to clients	The last responses sent to downstream clients (maximum of five unique clients), displayed in reverse chronological order. Each entry shows the IP address of the client and the timestamp sent.

Setting the system clock

In addition to supporting SNTP, the PowerConnect device also allows you to manually set the system clock. Using the **clock set** command starts the system clock with the time and date you specify. The time counter setting is retained across power cycles but is not automatically synchronized with an SNTP server.

NOTE
To synchronize the time counter with your SNTP server time, enter the **sntp sync** command from the Privileged EXEC level of the CLI.

NOTE
Unless you identify an SNTP server for the system time and date, system time is not guaranteed to be reliable.

For more details about SNTP, refer to "Specifying a Simple Network Time Protocol (SNTP) server" on page 90.

To set the system time and date to 10:15:05 on October 15, 2005, enter the following command.

```
NetIron# clock set 10:15:05 10-15-05
```

Syntax: [no] clock set <*hh:mm:ss*> <*mm-dd-yy*> | <*mm-dd-yyyy*>

By default, the PowerConnect router does not change the system time for daylight savings time. To enable daylight savings time, enter the following command.

```
NetIron# clock summer-time
```

Syntax: [no] clock summer-time

Although SNTP servers typically deliver the time and date in Greenwich Mean Time (GMT), you can configure the PowerConnect router to adjust the time for any one-hour offset from GMT or for one of the following U.S. time zones:

- US Pacific (default)
- Alaska
- Aleutian
- Arizona
- Central
- East-Indiana
- Eastern
- Hawaii
- Michigan
- Mountain
- Pacific
- Samoa

The default is US Pacific.

To change the time zone to Australian East Coast time (which is normally 10 hours ahead of GMT), enter the following command.

```
NetIron(config)# clock timezone gmt gmt+10
```

Syntax: [no] clock timezone gmt gmt | us <*time-zone*>

You can enter one of the following values for <*time-zone*>:

- US time zones (**us**): alaska, aleutian, arizona, central, east-indiana, eastern, hawaii, michigan, mountain, pacific, samoa.
- GMT time zones (**gmt**): gmt+12, gmt+11, gmt+10...gmt+01, gmt+00, gmt-01...gmt-10, gmt-11, gmt-12.

DST “change" notice for networks using US time zones

The new Daylight Saving Time (DST) change that went into effect on March 11, 2007 affects networks in the US time zones. Because of this change, your network clock might not be correct. If your network uses US time zones, and it needs to maintain the correct time, you must enable the following command.

```
NetIron(config)# clock timezone us pacific
```

Syntax: [no] clock timezone us {pacific | eastern | central | mountain}

NOTE
This command must be configured on every device that uses the US DST.

To verify the change, use the following command.

```
NetIron(config)# show clock
```

For more information, refer to the Marketing web site.

Creating a command alias

Use the **alias** command to create an alias for a command and to save that alias within the router's configuration.

To create the alias "shro" for the **show ip routes** command, use the following command.

```
NetIron(config)# alias shro = show ip routes
NetIron(config)# write memory
```

Syntax: [no] alias [<*name*> **=** <*command*> **]**

The <*name*> variable is the name that you want to assign to the alias.

The <*command*> variable is the syntax for the command you want to create an alias for.

The **write** memory command is used to save the alias within the configuration.

Removing an alias

You can remove an alias using the **no** version of the alias command as shown in the following.

```
NetIron(config)# no alias shro
```

Alternately, you can use the **unalias** command as shown in the following.

```
NetIron(config)# unalias shro
```

Syntax: [no] unalias

If the alias you try to remove does not exist, the following error will be displayed.

```
NetIron(config)# unalias wrs
Error: Alias wrs does not exist, unalias failed
```

Displaying a list of all configured alias

The following command allows you to display a list of all configured alias.

```
NetIron# alias
#alias
             savemem    write memory
             shro       show ip routes
```

Syntax: [no] alias

Limiting broadcast, multicast, or unknown unicast rates

The PowerConnect router can forward all traffic at wire speed. However, some third-party networking devices cannot handle high forwarding rates for broadcast, multicast, or unknown unicast packets.

The limits are individually configurable for broadcasts, multicasts, and unknown unicasts. You can configure limits globally to apply to each individual inbound interface module. The valid range is 1 – 4294967295 packets per second. The default is 0, which disables limiting.

Limiting broadcasts

To globally limit the number of broadcast packets a PowerConnect router forwards to 100,000 per second, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# broadcast limit 100000
NetIron(config)# write memory
```

Syntax: [no] broadcast limit <*number*>

Limiting multicasts

To globally limit the number of multicast packets a PowerConnect router forwards to 120,000 per second, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# multicast limit 120000
NetIron(config)# write memory
```

Syntax: [no] multicast limit <*number*>

NOTE
The multicast limit is configured at the global level, but the value you enter applies to each interface module (slot) installed on the device.

Limiting unknown unicasts

To globally limit the number of unknown unicast packets a PowerConnect router forwards to 110,000 per second, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# unknown-unicast limit 110000
NetIron(config)# write memory
```

Syntax: [no] unknown-unicast limit <*number*>

NOTE
Only the **unknown-unicast limit** is configured on the global level, but the value you enter applies to each interface module (slot) installed on the device.

Configuring CLI banners

The PowerConnect router can be configured to display a greeting message on users' terminals when they enter the Privileged EXEC CLI level or access the device through Telnet. In addition, a PowerConnect router can display a message on the Console when an incoming Telnet CLI session is detected.

Setting a message of the day banner

You can configure the PowerConnect router to display a message on a user's terminal when he or she establishes a Telnet CLI session. For example, to display the message "Welcome to PowerConnect!" when a Telnet CLI session is established, enter the following.

```
NetIron(config)# banner motd $(Press Return)
Enter TEXT message, End with the character '$'.
Welcome to PowerConnect! $
```

A delimiting character is established on the first line of the **banner motd** command. You begin and end the message with this delimiting character. The delimiting character can be any character except "(double-quotation mark) and cannot appear in the banner text. In this example, the delimiting character is $(dollar sign). The text in between the dollar signs is the contents of the banner. The banner text can be up to 2048 characters long and can consist of multiple lines. To remove the banner, enter the **no banner motd** command.

Syntax: [no] banner <*delimiting-character*> **| [motd** <*delimiting-character*>**]**

NOTE
The **banner** <*delimiting-character*> command is equivalent to the **banner motd** <*delimiting-character*> command.

NOTE
The size of the MOTD banner will be restricted (truncated) to 1850 characters when using an SSH client.

Setting a privileged EXEC CLI level banner

You can configure the PowerConnect router to display a message when a user enters the Privileged EXEC CLI level.

Example

```
NetIron(config)# banner exec_mode # (Press Return)
Enter TEXT message, End with the character '#'.
You are entering Privileged EXEC level
Don't foul anything up! #
```

As with the **banner motd** command, you begin and end the message with a delimiting character; in this example, the delimiting character is # (pound sign). To remove the banner, enter the **no banner exec_mode** command.

Syntax: [no] banner exec_mode <*delimiting-character*>

Displaying a message on the console when an incoming Telnet session Is detected

You can configure the PowerConnect router to display a message on the Console when a user establishes a Telnet session. This message indicates where the user is connecting from and displays a configurable text message.

Example

```
NetIron(config)# banner incoming $(Press Return)
Enter TEXT message, End with the character '$'.
Incoming Telnet Session! $
```

When a user connects to the CLI using Telnet, the following message appears on the Console.

```
Telnet from 209.157.22.63
Incoming Telnet Session!
```

Syntax: [no] banner incoming <*delimiting-character*>

To remove the banner, enter the **no banner incoming** command.

Configuring terminal display

You can configure and display the number of lines displayed on a terminal screen during the current CLI session.

The **terminal length** command allows you to determine how many lines will be displayed on the screen during the current CLI session. This command is useful when reading multiple lines of displayed information, especially those that do not fit on one screen.

To specify the maximum number of lines displayed on one page, enter a command such as the following.

```
NetIron(config)# terminal length 15
```

Syntax: [no] terminal length <*number-of-lines*>

The <*number-of-lines*> parameter indicates the maximum number of lines that will be displayed on a full screen of text during the current session. If the displayed information requires more than one page, the terminal pauses. Pressing the space bar displays the next page.

The default for <*number-of-lines*> is 24. Entering a value of 0 prevents the terminal from pausing between multiple output pages:

Checking the length of terminal displays

The **show terminal** command specifies the number of lines that will be displayed on the screen as specified by the **terminal length**, **page display**, and **skip-page-display** commands. It also shows if the **enable skip-page-display** command has been configured. The **enable skip-page-display** command allows you to use the skip-page-display to disable the configured page-display settings.

```
NetIron(config)# show terminal
Length: 24 lines
Page display mode (session): enabled
Page display mode (global): enabled
```

Syntax: show terminal

Enabling or disabling routing protocols

The Multi-Service IronWare supports the following protocols:

- BGP4

- DVMRP
- IP
- ISIS
- MPLS
- MSDP
- OSPF
- PIM
- RIP
- VRRP
- VRRPE

By default, IP routing is enabled on the PowerConnect router. All other protocols are disabled, so you must enable them to configure and use them.

To enable a protocol on a PowerConnect router, enter **router** at the global CONFIG level, followed by the protocol to be enabled. The following example shows how to enable OSPF.

```
NetIron(config)# router ospf
```

Syntax: [no] router bgp | dvmrp | ospf | pim | rip | vrrp | vrrpe

Displaying and modifying default settings for system parameters

The Multi-Service IronWare has default table sizes for the following parameters. The table sizes determine the maximum number of entries the tables can hold. You can adjust individual table sizes to accommodate your configuration needs:

- MAC address entries
- VLANs supported on a system
- Virtual interfaces
- Spanning tree instances
- RSTP instances
- IP cache size
- ARP entries
- IP routes
- IP ACL filter entries
- L2 ACL entries per ACL table
- Size for management port ACL
- IP subnets per port and per device
- IPv6 Multicast routes
- IPv6 PIM mcache
- Layer 4 sessions supported
- Number of VPLS's

- VPLS MAC entries
- IP VRF routes
- IPv6 cache
- IPv6 routes
- Number of tunnels
- Number of LAGs
- Configuration file size

The tables you can configure as well the defaults and valid ranges for each table differ depending on the PowerConnect router you are configuring.

NOTE
If you increase the number of subnet addresses you can configure on each port to a higher amount, you might also need to increase the total number of subnets that you can configure on the device.

NOTE
Changing the table size for a parameter reconfigures the device's memory. Whenever you reconfigure the memory on a PowerConnect router, you must save the change to the startup configuration file, then reload the software to place the change into effect.

To display the configurable tables and their defaults and maximum values, enter the following command at any level of the CLI.

FIGURE 1 Output for the Product Name

```
NetIron#show default values
sys log buffers:50         mac age time:300 sec       telnet sessions:5
ip arp age:10 min          bootp relay max hops:4     ip ttl:64 hops
ip addr per intf:24
when multicast enabled :
igmp group memb.: 260 sec  igmp query:       125 sec
when ospf enabled :
ospf dead:40 sec           ospf hello:10 sec          ospf retrans:5 sec
ospf transit delay:1 sec
when bgp enabled :
bgp local pref.:100        bgp keep alive:60 sec      bgp hold:180 sec
bgp metric:10              bgp local as:1             bgp cluster id:0
bgp ext. distance:20       bgp int. distance:200      bgp local distance:200
when IS-IS enabled :
isis hello interval:10 sec           isis hello multiplier:3
isis port metric:10                  isis priority:64
isis csnp-interval:10 sec            isis default-metric:10
isis distance:115                    isis lsp-gen-interval:10 sec
isis lsp-interval:33 msec            isis lsp-refresh-interval:900 sec
isis max-lsp-lifetime:1200 sec       isis maximum-paths:4
isis retransmit-interval:5 sec       isis spf-interval:5 sec
filter change update delay:10 sec
System Parameters    Default    Maximum    Current    Actual     Bootup Revertible
mac                    131072     2097152    2097152    2097152    2097152    Yes
vlan                   512        4095       4095       4095       4095       No
spanning-tree          32         128        128        128        128        No
rstp                   32         128        128        128        128        No
ip-arp                 8192       65536      65536      65536      65536      No
multicast-route (IPv6) 8192       153600     153600     153600     153600     Yes
pim-mcache (IPv6)      4096       4096       4096       4096       4096       Yes
ip-cache               204800     1048576    1048576    1048576    1048576    Yes
ip-route               204800     1048576    1048576    1048576    1048576    Yes
ip-subnet-port         24         128        128        128        128        No
virtual-interface      255        4095       4095       4095       4095       No
vpls-mac               8192       1000000    1000000    1000000    1000000    Yes
vpls-num               2048       16384      16384      16384      16384      No
session-limit          32768      163840     163840     163840     163840     Yes
ip-filter-sys          4096       40960      40960      40960      40960      No
mgmt-port-acl-size     20         100        100        100        100        No
l2-acl-table-entries   64         256        256        256        256        No
ipv6-cache             65536      245760     245760     245760     245760     Yes
ipv6-route             65536      245760     245760     245760     245760     Yes
ip-vrf-route           5120       262143     262143     262143     262143     Yes
receive-cam            1024       8192       8192       8192       8192       No
ip-tunnels              256       8192       8192       8192       8192        No
lsp-out-acl-cam        0          16384      16384      16384      16384      No
trunk-num              128        256        256        256        256        No
config-file-size       8388608    16777216   16777216   16777216   16777216   No
ifl-cam                0          81920      49152      49152      49152      No
ip-source-guard-cam    0          131072     30000      30000      30000      No
ipv4-mcast-cam         8192       65536      10000      10000      10000      No
ipv6-mcast-cam          2048       16384       3500       3500       3500
No
```

The following table describes the system-max values of the **show default values** command for NetIron MLX Series devices.

TABLE 14 Display of show default values for system parameters

This field...	Displays...
Default	The default value for the system-max element. This value is used in the following conditions: a)There is no system-max configured for the corresponding element. b) If the system-max element configuration is reverted at bootup time (if it is a revertible element).
Maximum	The maximum value that this element can be configured at.
Current	The most current configured value for the system-max element. If the system-max element is configured in the running system, then the value under this column will change to reflect this value. **NOTE:** The new value does not take affect until the node is reloaded.
Actual	The system-max value that is used by the target application of the running system. If system-max elements are reverted at bootup, then only the Actual column is affected. The Application is now using default values and will be displayed in the Actual column. Please refer to the example on the next page for more information. The Current and Bootup values are still configured on the system, and are not affected by the reversion of system-max elements at bootup.
Bootup	The system-max value that was read from the configuration when the system was booting up. If the read values are found to be acceptable, and not reverted, then the values in this column, and in the “Actual” column will have the same values. However, if the values were reverted during bootup, then the values are different for the “Revertible” elements.
Revertible	This column displays which corresponding system-max element is revertible or not. If “Yes” is displayed then the value is changed to a default value. If ‘No” is displayed then there no change to the value.

If system-max elements are reverted at bootup time, then the following message will display on the CLI.

```
NetIron#show default values
...

NOTE: All the Revertible Elements were Reverted During System Bringup.

System Parameters  Default    Maximum    Current    Actual     Bootup   Revertible
mac                  131072     2097152    2097152    131072     2097152    Yes
vlan                   512        4095       512        512        512        No
spanning-tree          32         128        32         32         32         No
rstp                   32         128        32         32         32         No
ip-arp                 8192       65536      65536      65536      65536      No
multicast-route (IPv6) 8192       153600     8192       8192       8192       Yes
pim-mcache (IPv6)      4096       4096       4096       4096       4096       Yes
ip-cache               204800     1048576    524288     204800     524288     Yes
...
```

Information for the configurable tables appears under the columns shown in bold type. To simplify configuration, the command parameter you enter to configure the table is used for the table name.

Example : To increase the capacity of the IP route table

```
NetIron(config)# system-max ip-route 120000
NetIron(config)# write memory
NetIron(config)# exit
NetIron# reload
```

NOTE
If you enter a value that is not within the valid range of values, the CLI will display the valid range for you.

To increase the number of IP subnet interfaces you can configure on each port on a PowerConnect router to 64, enter the following commands.

```
NetIron(config)# system-max ip-subnet-port 64
NetIron(config)# write memory
NetIron(config)# exit
NetIron# reload
```

Syntax: [no] system-max ip-subnet-port <*num*>

The <*num*> parameter specifies the maximum number of subnet addresses per port. The minimum, maximum and default values for this parameter are described in Table 15.

To increase the size of the IP route table, enter the following command.

```
PowerConnect(config)# system-max ip-static-route 400000
```

Syntax: [no] system-max ip-route <*num*>

The minimum, maximum and default values for this parameter are described in Table 15.

NOTE
You must reload the software for the change to take effect.

Enabling or disabling layer 2 switching

By default, PowerConnect devices supports routing over layer 2 switching. You can enable layer 2 switching globally or on individual port using the **no route-only** command.

The **no route-only** and **route-only** commands prompts you for whether or not you want to change the "route-only" behavior. You must enter **y** if you want to proceed or **n** if you do not. The prompt is displayed as shown in the following examples of the **no route-only** and **route-only** commands.

To enable Layer 2 switching globally, enter the following.

```
NetIron(config)# no route-only
This will change the route-only behavior at the global level.
Are you sure? (enter 'y' or 'n'): y
Global 'route-only' committed.
```

To globally disable Layer 2 switching on a PowerConnect router and return to the default (route-only) condition, enter commands such as the following:

```
NetIron(config)# route-only
This will change the route-only behavior at the global level.
Are you sure? (enter 'y' or 'n'): y
Global 'no route-only committed.
```

Syntax: [no] route-only

To enable Layer 2 switching only on a specific interface, go to the Interface configuration level for that interface, and add the **no route-only** command. The following commands show how to enable Layer 2 switching on port 3/2.

```
NetIron(config)# interface ethernet 3/2
NetIron(config-if-e10000-3/2)# no route-only
```

Syntax: [no] route-only

To re-enable the default **route-only** condition on port 3/2, enter the **route-only** command as shown.

```
NetIron(config-if-e10000-3/2)# route-only
```

The default route-only condition, unknown unicast packets are not sent to the CPU and are dropped locally by the hardware.

Configuring static MAC addresses

You can assign static MAC addresses to ports of a PowerConnect router.

You can manually input the MAC address of a device to prevent it from being aged out of the system address table, to prevent traffic for a specific device, such as a server, from flooding the network with traffic when it is down, and to assign higher priorities to specific MAC addresses.

Static MAC addresses are configured within a specified VLAN including the default VLAN 1. Optionally you can specify a port priority (QoS).

The default and maximum configurable MAC table sizes can differ depending on the device. To determine the default and maximum MAC table sizes for your device, display the system parameter values. Refer to "Displaying and modifying default settings for system parameters" on page 100.

The ability of the CAM to store depends on the following:

- The number of source MAC address being learned by the CAM.
- The number of destination MAC addresses being forwarded by the CAM
- The distribution of the MAC address entries across ports. For example, it one port is learning all the source MAC addresses, the available of the CAM for that port will be depleted.

Example

In the following example, a static MAC address of 1145.5563.67FF with a priority of 7 is assigned to port 2 of module 1 in VLAN 200.

```
NetIron(config)# vlan 200
NetIron(config)# static-mac-address 1145.5563.67FF e 1/2 priority 7
```

Syntax: [no] static-mac-address <*mac-addr*> **ethernet** <*portnum*> **[priority** <*number*>**]**

The <*mac-addr*> variable specifies the MAC address that you assigning.

The **portnum** variable specifies the Ethernet port that the MAC address is being assigned to.

Using the **priority** option, you can assign a value to the <*number*> variable of 0 – 7

Changing the MAC age time

The MAC age time sets the aging period for ports on the device, defining how long (how many seconds) a port address remains active in the address table.

To change the aging period for MAC addresses from the default of 300 seconds to 600 seconds.

```
NetIron(config)# mac-age-time 600
```

Syntax: [no] mac-age-time <*age-time*>

The <*age-time*> can be 0 or a number from 67 – 65535. The zero results in no address aging. The default is 300 (seconds).

Configuring static ARP entries

When you create a static ARP entry, the PowerConnect router automatically creates a static MAC entry.

NOTE
To delete the static MAC entry, you must delete the static ARP entry first.

Configuring system max values

The system max values for the several system parameters of the NetIron MLX Series devices are described in Table 15

TABLE 15 System max values for NetIron MLX Series routers

Parameter	Minimum value for MLXe	Maximum value for MLXe	Default value for MLXe
config-file-size	2097152	16777216	8388608
gre-tunnels	1	8192	256
hw-flooding	0	4095	0
ifl-cam	0	81920	0
ip-arp	2048	65536	8192
ip-cache	8192	524288	102400
ip-filter-system	1024	40960	4096
ip-route	4096	524288	102400
ipv4-mcast-cam	0	32768	4096
ip-subnet-port	24	128	24
ip-vrf-route	128	262143	5120
ipv6-cache	8192	102400	32768
ipv6-mcast-cam	0	8192	1024
ipv6-route	4096	114688	32768
l2-acl-table-entries	64	256	64

TABLE 15 System max values for NetIron MLX Series routers (Continued)

Parameter	Minimum value for MLXe	Maximum value for MLXe	Default value for MLXe
mac	4000	1048576	32768
mgmt-port-acl-size	1	100	20
receive-cam	512	8192	1024
rstp	1	128	32
session-limit	1024	40960	8192
spanning-tree	1	128	32
virtual-interface	40	4095	255
vlan	2	4095	512
vpls-mac	32	262144	2048
vpls-num	512	4096	512

To configure system-max values, use the following command.

Syntax: [no] system-max config-file-size | gre-tunnels | hw-flooding | ip-arp | ip-cache |ip-filter-sys | ip-route | ip-static-arp | ipv4-mcast-cam | ip-subnet-port | ip-tunnels| ip-vrf| ip-vrf-route | ipv6-cache | ipv6-mcast-cam | ipv6-route | l2-acl-table-entries |ifl-cam | mac | mgmt-port-acl-size | receive-cam | rstp | session-limit | spanning-tree | trunk-num | virtual-interface | vlan | vpls-mac | vpls-num

The **gre-tunnels** parameter sets the maximum number of GRE tunnels.
For minimum, maximum and default values for this parameter refer to Table 15.

The **config-file-size** parameter sets the allowed running and startup-config file sizes. Refer to the appropriate table for your platform. For minimum, maximum and default values for this parameter refer to Table 15.

The **hw-flooding** parameter sets the maximum number of hardware flooding entries.
For minimum, maximum and default values for this parameter refer to Table 15. T

NOTE
The dynamic growth of the protocol and flooding sub-partitions in the L2 CAM to grant them a higher priority has been enhanced. Because of this enhancement, the **system-max hw-flooding** command is no longer required and has been retired from the software. If you already have the **system-max hw-flooding** command in your configuration, it will automatically be upgraded to the current functionality and the command will be deleted from the configuration file.

The **ifl-cam** parameter sets the maximum number of Internal Forwarding Lookup Indentifiers. These are used when configuring a Local VLL for Dual Tagging. The default value for the **ifl-cam** parameter is 8K. The maximum values for this parameter are different depending on which CAM partition you have configured on your system. For minimum, maximum and default values by CAM partition for this parameter, refer to Table 16.

The **ip-arp** parameter sets the maximum number of ARP entries.
For minimum, maximum and default values for this parameter refer to Table 15.

The **ip-cache** parameter sets the maximum size of the IP cache.
For minimum, maximum and default values for this parameter refer to Table 15.

The **ip-filter-sys** parameter sets the maximum number of IP ACL entries.
For minimum, maximum and default values for this parameter refer to Table 15.

The **ip-route** parameter sets the maximum number of IP Route entries.
For minimum, maximum and default values for this parameter refer to Table 15.

NOTE
There is no need to configure a system-max value for static ARP entries.

The **ip-static-arp** parameter sets the maximum number of static ARP entries.
For minimum, maximum and default values for this parameter refer to Table 15.

The **ipv4-mcast-cam** parameter allows you to configure the maximum CAM size for an IPv4 multicast group. For minimum, maximum and default values for this parameter refer to Table 4.5. To configure the CAM size of an IPv4 multicast group, refer to "Configuring CAM size for an IPv4 multicast group" on page 109.

The **ip-subnet-port** parameter sets the maximum number of IP subnets per port.
For minimum, maximum and default values for this parameter refer to Table 15.

The **ip-vrf-route** parameter sets the maximum number of IP VRF routes per VRF instance.
For minimum, maximum and default values for this parameter refer to Table 15.

The **ipv6-cache** parameter sets the maximum size of the IPv6 cache.
For minimum, maximum and default values for this parameter refer to Table 15.

The **ipv6-mcast-cam** parameter allows you to configure the maximum CAM size for an IPv6 multicast group. For minimum, maximum and default values for this parameter refer to Table 14. To configure the CAM size of an IPv6 multicast group, refer to "Configuring CAM size for an IPv6 multicast group" on page 110.

The **ipv6-route** parameter sets the maximum number of IPv6 routes.
For minimum, maximum and default values for this parameter refer to Table 15.

NOTE
The **system-max ipv6-route** command can be configured with a maximum value of 114688 on the Product Name, however the PowerConnect system will only support a maximum value of 114687 for ipv6 routes.

The **l2-acl-table-entries** parameter sets the maximum number of layer-2 ACL entries per ACL table.
For minimum, maximum and default values for this parameter refer to Table 15.

The **mac** parameter sets the maximum number of MAC entries.
For minimum, maximum and default values for this parameter refer to Table 15.

The **mgmt-port-acl-size** parameter sets the maximum size for a management port ACL.
For minimum, maximum and default values for this parameter refer to Table 15.

The **receive-cam** parameter sets the maximum number of IP Receive ACL software CAM entries.
For minimum, maximum and default values for this parameter refer to Table 15.

The **rstp** parameter sets the maximum number of RSTP instances.
For minimum, maximum and default values for this parameter refer to Table 15.

The **session-limit** parameter sets the maximum number of sessions.
For minimum, maximum and default values for this parameter refer to Table 15.

The **spanning-tree** parameter sets the maximum number of spanning-tree instances. For minimum, maximum and default values for this parameter refer to Table 15.

The **virtual-interface** parameter sets the maximum number of virtual interfaces. For minimum, maximum and default values for this parameter refer to Table 15.

The **vlan** parameter sets the maximum number of VLANs. For minimum, maximum and default values for this parameter refer to Table 15.

The **vpls-mac** parameter sets the maximum number of VPLS MAC Entries. For minimum, maximum and default values for this parameter refer to Table 15.

The **vpls-num** parameter sets the maximum number of Virtual Private LAN Services (VPLS). For minimum, maximum and default values for this parameter refer to Table 15

TABLE 16 System max **ifl-cam** values available by CAM profile on PowerConnect B-MLXe

CAM profile	Minimum value	Maximum value	Default value
Default	0	57344	8192
ipv4	0	114688	8192
ipv6	0	16384	8192
l2-metro	0	114688	8192
mpls-l3vpn	0	114688	8192
mpls-vpls	0	114688	8192
multi-service	0	49152	8192
multi-service-2	0	81920	8192
vpn-vpls	0	114688	8192
ipv4-vpn	0	114688	8192
l2-metro-2	0	114688	8192
mpls-l3vpn-2	0	114688	8192
mpls-vpls-2	0	114688	8192
ipv4-ipv6	0	114688	8192
ipv4-vpls	0	114688	8192
ipv4-ipv6-2	0	81920	8192

Configuring CAM size for an IPv4 multicast group

To configure the CAM size of an IPv4 multicast group, enter the following command.

```
NetIron(config)# system-max ipv4-mcast-cam
```

Syntax: [no] system-max [ipv4-mcast-cam]

By default, no system-max parameter is configured.

The **ipv4-mcast-cam** parameter allows you to specify the maximum CAM size you want for an IPv4 multicast group.

The **decimal** parameter specifies the range that is supported for configuring the CAM size. On the PowerConnect B-MLXe, the minimum value supported is 0, and the maximum value supported is 32768. The default value is 4096.

Upon configuration, the NetIron system will verify the input value with the amount of CAM resources that are available. If the PowerConnect system is unable to allocate requested space, it will display the following error message on the Product Name.

```
NetIron(config)# system-max ipv4-mcast-cam 32768
Error - IPV4 Multicast CAM (32768) exceeding available CAM resources
Total IPv4 ACL CAM:             49152(Raw Size)
IPv4 Receive ACL CAM:           2048(Raw Size)
IPv4 Source Guard CAM:          0(Raw Size)
Reserved IPv4 Rule ACL CAM:     1024(Raw Size)
Available IPv4 Multicast CAM:   46080(Raw Size)  23040(User Size)
```

If there is not enough CAM resources available to change the cam-partition profile from IPv4 to IPv6, the following message is displayed.

```
NetIron(config)# cam-partition profile ipv4-ipv6
Error - IPv4 Receive ACL CAM (1024) exceeding available CAM resources
Total IPv4 ACL CAM:                 49152(Raw Size)
    IPv4 Multicast CAM:             65536(Raw Size)
    IPv4 Source Guard CAM:          0(Raw Size)
    Reserved IPv4 Rule ACL CAM:     1024(Raw Size)
    Available IPv4 Receive ACL CAM: 0(Raw Size) 0(User Size)
Error - Failed to select this CAM profile
```

This error message is also displayed on the Product Name.

After you issue the system-max command, with ipv4-mcast-cam parameter included, additional information will display on the Product Name as shown in the following example.

```
NetIron(config)#system-max ipv4-mcast-cam 60000
Reload required. Please write memory and then reload or power cycle.
Failure to reload could cause system instability on failover.
Newly configured system-max will not take effect during hitless-reload.
```

NOTE
You must write this command to memory and perform a system reload for this command to take effect.

Configuring CAM size for an IPv6 multicast group

To configure the CAM size of an IPv6 multicast group, enter the following command.

```
NetIron(config)# system-max ipv6-mcast-cam
```

Syntax: [no] system-max [ipv6-mcast-cam]

By default, no system-max parameter is configured.

The **ipv6-mcast-cam** parameter allows you to specify the maximum CAM size you want for an IPv6 multicast group.

The **decimal** parameter specifies the range that is supported for configuring the CAM size. On the Product Name, the minimum value supported is 0, and the maximum value supported is 8192. The default value is 1024.

Upon configuration, the NetIron system will verify the input value with the amount of CAM resources that are available. If the PowerConnect system is unable to allocate requested space, it will display the following error messages on the Product Name:

On the Product Name.

```
NetIron MLX(config)#system-max ipv6-mcast-cam 8000
Error - IPV6 Multicast CAM (8000) exceeding available CAM resources
Total IPv6 ACL CAM: 16384(Raw Size)
Reserved IPv6 Rule ACL CAM: 1024(Raw Size)
Available IPv6 Multicast CAM: 15360(Raw Size) 1920(User Size)
```

After you issue the system-max command, with ipv6-mcast-cam parameter included, additional information will display on the Product Name as shown in the following example.

```
NetIron MLX(config)#system-max ipv6-mcast-cam 1000
Reload required. Please write memory and then reload or power cycle.
Failure to reload could cause system instability on failover.
Newly configured system-max will not take effect during hitless-reload.
```

NOTE
You must write this command to memory and perform a system reload for this command to take effect.

Configuring profiles with a zero-size IPv4 or IPv6 ACL

When a profile is configured to a zero-size IPv4 or IPv6 ACL, the minimum value of 512 for IPv4, and the minimum value of 128 for IPv6 is disabled.

If the system-max value for multicast is configured, and you select a value of 0 for IPv4 or IPV6 ACL, the system-max value will be ignored. There is minimal checking for errors. The following warning message is displayed.

```
NetIron MLXe-MLX(config)#cam-partition profile mpls-vpls-2
Warning - Changing to a profile with zero Ipv6 ACL CAM size, ignoring system-max
value check and minimum-guarantee checks.
Reload required. Please write memory and then reload or power cycle.
Failure to reload could cause system instability on failover.
Newly configured system-max will not take effect during hitless-reload.
```

NOTE
You must write this command to memory and perform a system reload for this command to take effect.

The system-max value is retained in the configuration so that it can be reused later when you want to change the CAM profile.

Maintaining system-max configuration with available system resources

When system-max values are configured, the PowerConnect system checks for available system resources. The system resources are required in order to maintain dynamic memory allocation. System-max values are checked at the configuration time, and at the bootup time. If there are insufficient system resources available on the Management Module, this will cause the configuration to be rejected during card bootup. On the Interface Module, insufficient system resources will lead to failure in booting up the card.

Configuration time

When system-max values are configured, the Management Module calculates the memory required to accept the value. The resulting value is checked against the Known-Available-Memory value, and calculated against the Highest Required Memory value for both the Management Module and the Interface Module.

The Known-Available-Memory is a value with the Lowest Supported Available Memory on a node. For example, if a node can accept a 1 Gigabyte LP, and a 512 MB LP, then the 512 MB LP will be used. The Highest Required Memory is a value with most amount of memory available on a node. For example, if a node has both 2 PPCR LP, and 1 PPCR LP, then the 2 PPCR LP will be used.

If the new system-max value is accepted, then the configuration will also be accepted. The following information will display.

```
NetIron(config)#system-max mac 4000
Reload required. Please write memory and then reload or power cycle.
Failure to reload could cause system instability on failover.
Newly configured system-max will not take effect during hitless-reload.
```

NOTE
You must write this command to memory and perform a system reload for this command to take effect.

If the new system-max value is not accepted, then the configuration is rejected. The following error message is printed on the console.

```
NetIron(config)# system-max ipv4 10000
ERROR: Configured System-max value cannot be accommodated.
```

Bootup time

At bootup time, the Management Module will repeat the same process as done in the Configuration time. The Management Module calculates the memory required to accept the system-max configuration. The resulting value is checked against the Known-Available-Memory value for both the Management Module and the Interface Module.

After the new system-max value is configured, there are three possible configuration outcomes. The three possible configuration outcomes are described below.

1. The configuration can be accommodated, but leaves only 10% of Available Memory

   In this configuration, a check is made against 90% of Available Memory. If the difference between the Required Available Memory and the Available Memory is less than 10% of Available Memory, then the configuration is accepted. The following warning message is displayed on the console if it affects the Management Module or Interface Module.

   The following warning message is displayed on the Management Module.

   ```
   WARNING: Configured System-max Leaves less than 10 percent Free Memory
   Available on MP.
   ```

   The following warning message is displayed on the Interface Module.

   ```
   WARNING: Configured System-max Leaves less than 10 percent Free Memory
   Available on LP.
   ```

   A syslog message showing the required memory versus the available memory is generated, and a similar warning message is displayed on the Management Module and Interface Module as shown in the following example.

```
NetIron# show log
…
Jan 17 22:55:55:N: WARN: Configured System-max Leaves less than 10 percent Free
Memory Available on MP (162529285 req vs 1625292800 available)
Jan 17 22:55:55:N: WARN: Configured System-max Leaves less than 10 percent Free
Memory Available on LP (162529285 req vs 1625292800 available)
```

NOTE
When the system is booted up again, the percent of free memory is discretionary and is only an estimate.

NOTE
Even if all elements are configured with the maximum allowed value, you may not see the reversion of system-max values that occur on any given Interface Module.

NOTE
Notifications and traps are sent with the same message.

2. The configuration can be easily accommodated.

   In this configuration, the Management Module continues to use the configured system-max value, and send the same value to the installed Interface Modules.

3. The configuration cannot be accommodated.

   If the configured system-max value cannot be used, the Management Module will locate the elements that can be reverted to a default value. These system-max elements will revert to a default value, and the following message will display on the console.

   ```
   WARN: Configured System-max cannot be accommodated. Resetting revertible
   elements to default values.
   ```

A syslog message is generated, and a similar warning message is displayed on the Management Module and Interface Module as shown in the following example.

```
NetIron# show log
…
Jan 17 22:55:55:N: WARN: Configured System-max cannot be accommodated on MP
(1625292801 req vs 1625292800 available). Resetting revertible elements to
default
values.
Jan 17 22:55:55:N: WARN: Configured System-max cannot be accommodated on LP
(1625292801 req vs 1625292800 available). Resetting revertible elements to
default
values.
```

NOTE
Once the system-max have been reverted, a user might not be able to configure any system-max until configuration for some or all of the revertible system-max elements is removed using "no system-max" CLI.

NOTE
Notifications and traps are sent with the same message.

The following tables show which elements are revertible (Yes or No) in each element category.

L2 elements

TABLE 17 L2 elements

L2 elements	Revertible: yes or no
Mac	yes
Vlan	no
Spanning-tree	no
Rstp	no

L3 elements

TABLE 18 L3 elements

L3 elements	Revertible: yes or no
Arp	no
multicast-route (for v6 only)	yes
pim-mcache	yes
ip-cache	yes
ip-route	yes
ip-subnet-port	no
virtual-interface	no

VPLS elements

TABLE 19 VPLS elements

VPLS elements	Revertible: yes or no
vpls-mac (MAX_VPLS_MAC_INDEX)	yes
vpls-num (MAX_VPLS_NUM_INDEX)	no

Miscellaneous elements

TABLE 20 Miscellaneous elements

Miscellaneous elements	Revertible: yes or no
session-limit	yes
ip-filter-sys	no
mgmt-port-acl-size	no
l2-acl-table-entries	no
ipv6-cache	yes
ipv6-route	yes
IPVRF MAX ROUTES	yes
mgmt-port-acl-size	no
receive-cam	no
IPGRE	no
LSP_ACL	no
SERVICE_LOOKUP	no
IP_SRC_GUARD_CAM	no
IPv4 MCAST CAM	no
IPv6 MCAST CAM	no
SERVER_TRUNKS	no
CONFIG_FILE_SIZE	no

Monitoring dynamic memory allocation

After a configured system-max value is accepted, it is possible that the dynamic memory allocation may fail in a running system. To monitor the amount of available memory on the Management Module and the Interface Module, a timer will check the memory every 10 seconds. If the available memory falls below 5 percent of the total installed memory, the timer will log the following warning message.

```
NetIron# show log
…
Jan 17 22:55:55:N: WARN: Current Total Free Memory on MP is below 5 percent of
Installed Memory.
…
Jan 17 23:53:55:N: WARN: Current Total Free Memory on LP 8 is below 5 percent of
Installed Memory.
```

The warning message is displayed at a frequency of 1 log per 5 minutes.

NOTE
Notifications and traps are sent.

When the memory allocation fails, an alert message is logged immediately. The alert message is displayed at a frequency of 1 log per 5 minutes. The following example below displays an alert message on the Management Module and the Interface Module.

```
NetIron# show log
…
Jan 17 22:55:55:A: ALERT: Failed to allocate memory on MP
…
Jan 17 23:52:55:A: ALERT: Failed to allocate memory on LP 8
…
```

The NULL value is returned to the calling routine. The calling routine will decide how to proceed after the memory allocation fails.

NOTE
Notifications and traps are sent.

At any time, you can display the status of all recorded memory that is available on the Management Module by entering the **show memory** command. The amount of available memory is displayed in percentage values.The following example displays a show memory output on a Management Module.

```
NetIron# show memory
====================================================================
NetIron MLX active MP slot 33:
Total SDRAM         :  1073741824 bytes
Available Memory    :   720007168 bytes
Available Memory (%):          66 percent
Free Physical Pages :      160176 pages

Malloc statistics: total  150898213
OS malloc count:            271413
OS malloc fail:                  0
OS free count:              266179
OS free fail:                    0
diff:                         5234
====================================================================
NetIron MLX standby MP slot 34:
Total SDRAM         :  1073741824 bytes
Available Memory    :   704876544 bytes
Available Memory (%):          65 percent
Free Physical Pages :      167235 pages
```

Switch fabric fault monitoring

With this feature, you can display information about the current status of links between the switch fabric modules (SFM) and interface modules in a PowerConnect B-MLXe chassis. This feature also provides log messages to the console when there is a change in the “UP” or “DOWN” status of links to the SFM and when an individual fabric element (FE) cannot be accessed by the management module. The router can also be configured to automatically shut down an SFM when failure is detected. The following sections describe the capabilities of this feature.

Displaying switch fabric information

You can display information about the current status of links between the SFMs and interface modules in a PowerConnect B-MLXe chassis using the following command. Each line represents a link between an SFM and an interface module (LP).

```
NetIron#show sfm-links all
 SFM#/FE# | FE link#  | LP#/TM# | TM link#  | link state
----------+-----------+---------+-----------+---------
  2 / 1   |    32     | 3 / 1   |    13     |      UP
  2 / 1   |    31     | 3 / 2   |    01     |      UP
  2 / 1   |    11     | 3 / 1   |    01     |      UP
  2 / 1   |    12     | 3 / 2   |    13     |      UP

  2 / 3   |    32     | 3 / 1   |    19     |      UP
  2 / 3   |    31     | 3 / 2   |    07     |      UP
  2 / 3   |    11     | 3 / 1   |    07     |      UP
  2 / 3   |    12     | 3 / 2   |    19     |      UP

  3 / 1   |    32     | 3 / 1   |    16     |      UP
  3 / 1   |    31     | 3 / 2   |    04     |      UP
  3 / 1   |    11     | 3 / 1   |    04     |      UP
  3 / 1   |    12     | 3 / 2   |    16     |      UP

  3 / 3   |    32     | 3 / 1   |    22     |      UP
  3 / 3   |    31     | 3 / 2   |    10     |      UP
  3 / 3   |    11     | 3 / 1   |    10     |      UP
  3 / 3   |    12     | 3 / 2   |    22     |      UP

WARN: LP 3 has 8 links up, less than minimum to guarantee line rate traffic
forwarding
```

Syntax: show sfm-links <*sfm-number*> **| all [errors]**

The <*sfm-number*> variable specifies an SFM that you want to display link information for.

The **all** option displays link information for all SFMs in the chassis.

The **errors** option only displays information for SFM links that are in the DOWN state.

The output of this command can also be filtered using an output modifier. To use an output modifier, type a vertical bar (**|**) followed by a space and one of the following parameters:

- **begin** - begin output with the first matching line
- **exclude** - exclude matching lines from the output
- **include** - include only matching lines in the output

A warning statement is sent if the number of operational links falls below the minimum threshold. This warning is displayed to warn users that the line rate traffic will not be maintained.

The **show sfm-links** command displays the following information.

TABLE 21 CLI display of SFM link information

This field...	Displays...
SFM#	The switch fabric module number.
FE#	The FE number.
FE link#	The number of the interconnect between the SFM and the FE.
LP#	The slot number where the Interface module (LP) is installed.
TM#	The number of the traffic manager used in the link.
TM link#	The link number on the traffic manager.
link state	The link state is either: UP – In an operating condition DOWN – In a non-operational condition

Displaying switch fabric module information

To display the state of all switch fabric modules in the chassis, enter the following command at any level of the CLI.

```
NetIron> show module
M1 (upper): NI-MLX-MR Management Module Active
M2 (lower): NI-MLX-MR Management Module Standby (Ready State)
F1: NI-X-SF Switch Fabric Module Powered off (By Health Monitoring)
F2:
F3:
F4: NI-X-HSF Switch Fabric Module Active
...
```

Syntax: show module

The **show module** command displays the modules currently connected to the chassis and their state. For switch fabric modules, the command shows “Active” if the module is operational or “Powered off” and the reason for the shutdown.

Powering a switch fabric link on or off manually

To manually power on a switch fabric link, use a command such as the following.

```
NetIron# power-on snm-link sfm 3 fe 3 link 37
```

To manually power off a switch fabric link, use a command such as the following.

```
NetIron# power-off snm-link sfm 3 fe 3 link 37
```

Syntax: [no] power-on snm-link sfm <*sfm-number*> **fe** <*fe-number*> <*link-number*>

Syntax: [no] power-off snm-link sfm <*sfm-number*> **fe** <*fe-number*> <*link-number*>

Powering a switch fabric module off automatically on failure

To configure the router to automatically power off a switch fabric module (SFM) or high speed switch fabric module (hSFM) on which an access error has been detected, enter the following command at the CONFIG level of the CLI.

```
NetIron(config)# system-init fabric-failure-detection
```

Syntax: [no] system-init fabric-failure-detection

NOTE
You must restart the router for automatic SFM shutdown to take effect.

Once you have configured automatic SFM shutdown on the router and restarted it, the management module will automatically detect access failure (see "Access failure messages" on page 120) and shut down the unresponsive SFM. You can restart the SFM at any time (manually, by removing and re-inserting the module, or by initiating a system restart), but if another access error is detected, the management module will shut the SFM down again. If an SFM is automatically powered down, SFM power-off status (and the associated reason) are synced to the standby management module, and in the event of failover the standby module will keep the faulty SFM powered off.

Switch fabric log messages

Information about the state of each switch fabric module and whether it can be accessed by the Management Module is also provided in the form of syslog messages.

Link up/down messages

The Switch Fabric modules (SFM) in an PowerConnect chassis send a log message when they first become operational or when they change state between "UP" and "DOWN". The following is an example of the message sent when a link first becomes operational (UP) or when it changes state from non-operational (DOWN) to operational (UP).

```
Apr  6 10:57:20:E: Fabric Monitoring Link Up : SFM 3/FE 3/Link 37, LP 5/TM 1
```

The following is an example of the message sent when a link is detected going from operational (UP) to non-operational (DOWN).

```
Apr  6 10:56:00:E: Fabric Monitoring Link Down : SFM 3/FE 3/Link 37, LP 5/TM 1
```

Once a link had been detected as going down, it is automatically shut down by the Multi-Service IronWare software. The following is an example of the message sent when a link is either brought down automatically or manually using the command described in "Powering a switch fabric link on or off manually" on page 118.

```
Apr  6 10:56:00:E: Fabric Monitoring Link Admin Shut Down : SFM 3/FE 3/Link 37, LP
5/TM 1
```

This contents of the message are defined as described in the following.

`Apr  6 10:57:20:` – The time that the link changed state.

`Fabric Monitoring Link Up` – the link went "UP"
`Fabric Monitoring Link Down` – the link went "DOWN"

`SFM 3` – The switch fabric module (SFM) number

`FE 3` – The Fabric Element number

`Link 37` – The number of the interconnect between the SFM and the FE

`LP 5` – The slot number where the Interface Module (LP) is installed.

`TM 1` – The number of the traffic manager (TM) used in the link.

Access failure messages

The management module attempts to access each fabric element for every poll period (1 second by default). If the number of access failures in a poll window (default 10 seconds) exceeds the threshold (3 by default), the management module sends a log message similar to the following:

```
Apr 6 20:33:57:A:System: Health Monitoring: FE access failure detected on SFM 2/FE
1
```

The contents of the message are defined as described in the following.

`Apr 6 20:33:57:` – the time at which the error threshold was exceeded

`FE access failure detected` – the management module failed to access the specified FE

`SFM 2` – the switch fabric module (SFM) number

`FE 1`– the Fabric Element (FE) number

If the router has been configured to shut down a switch fabric module when failure is detected (see"Powering a switch fabric module off automatically on failure" on page 119), the management module will shut down the failed switch fabric module, then send a log message similar to the following:

```
Oct 4 20:33:57:A:System: Health Monitoring: Switch fabric 2 powered off due to
failure detection
```

The message above indicates that a failure was detected in attempting to access switch fabric module 2, and the module was powered off on October 4th at 20:33:57.

Switch fabric utilization monitoring

With this feature, you can monitor the percentage of the total bandwidth used on the SFM for the timing intervals of 1 sec, 5 sec, 1 min, and 5 min. For example, to display bandwidth usage on all SFMs on the device, enter the following command.

```
NetIron#show sfm-utilization all
SFM#2
----------+----------+---------+----------+---------
last 1 second utilization = 0.4%
last 5 seconds utilization = 0.3%
last 1 minute utilization = 0.1%
last 5 minutes utilization = 0.0%
SFM#3
----------+----------+---------+----------+---------
last 1 second utilization = 0.4%
last 5 seconds utilization = 0.4%
last 1 minute utilization = 0.1%
last 5 minutes utilization = 0.0%
```

To display bandwidth usage on one SFM, enter the following command.

```
NetIron#show sfm-utilization 2
SFM#2
----------+----------+---------+----------+---------
last 1 second  utilization = 0.4%
last 5 seconds utilization = 0.3%
last 1 minute  utilization = 0.1%
last 5 minutes utilization = 0.0%
```

Syntax: show sfm-utilization [<*sfm-number*> **| all]**

The <*sfm-number*> variable specifies an SFM that you want to utilization information for.

The **all** option displays utilization information for all SFMs in the chassis.

Verifying an image checksum

Use the **image-checksum** command to verify the checksum of the application, boot, or monitor images that are saved in code flash and PCMCIA cards.

To check a monitor image, use the following command.

```
NetIron# image-checksum monitor
OK
```

Syntax: [no] image-checksum <*file-name*>

The <*file-name*> variable specifies the image file that you want to verify the checksum for.

The following output can be generated by this command

TABLE 22 Output from image-checksum command

Output	Description
File not found	The router failed to locate the specified file.
Failed to read file	The router failed to obtain the file length from the file system.
Not an image file	The specified file is not an image file.
File read failed	The specified file's actual length is different form the file length stored in the file system.
Checksum failed	The image has a checksum error.
OK	The checksum has been verified for the specified image file.

Displaying information for an interface for an Ethernet port

To display information for a show interface for an ethernet port, enter the following command at any CLI level.

```
NetIron# show interface ethernet 9/1
GigabitEthernet2/3 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 0012.f298.4900 (bia 0012.f298.492a)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
 Member of VLAN 1 (untagged), 5 L2 VLANS (tagged), port is in dual mode (default
vlan), port state is Forwarding
  STP configured to ON, Priority is level0, flow control enabled
  Priority force disabled, Drop precedence level 0, Drop precedence force disabled
  arp-inspection-trust configured to OFF
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  Port name is ->7.blade1.shelf1.access.aprd
  MTU 1544 bytes, encapsulation ethernet
  300 second input rate: 1509512 bits/sec, 713 packets/sec, 0.15% utilization
  300 second output rate: 1992071 bits/sec, 751 packets/sec, 0.20% utilization
  712896623 packets input, 204984611768 bytes, 0 no buffer
  Received 1315502 broadcasts, 53313 multicasts, 711527808 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 29433839 giants
  NP received 712896745 packets, Sent to TM 712839428 packets
  NP Ingress dropped 57317 packets
  796106728 packets output, 366570033985 bytes, 0 underruns
  Transmitted 2045784 broadcasts, 32330616 multicasts, 761730328 unicasts
  0 output errors, 0 collisions
  NP transmitted 796106833 packets, Received from TM 796534170 packets
```

Syntax: show interface [ethernet <*slot-port*> **[to** <*slot-port*> **]]**

You can display information for all ports in a router by using the **show interface** command without options, or use the **ethernet** <*slot-port*> option to limit the display to a single port, or add the **to** <*slot-port*> option for a range of ports.

Displaying the full port name for an Ethernet interface

To display the full port name for an ethernet interface using the CLI, enter the following command..

```
NetIron# show interface brief wide
Port  Link L2 State  Speed Tag MAC            Name
2/1   Up   Forward   10G   No  0012.f2f7.0230 port-connected-to-chicago
2/2   DisabNone      None  No  0012.f2f7.0231
2/3   DisabNone      None  No  0012.f2f7.0232
2/4   DisabNone      None  No  0012.f2f7.0233

Port  Link L2 State  Speed Tag MAC            Name
mgmt1 Up   Forward   100M  Yes 0012.f2f7.0200

Port  Link L2 State  Speed Tag MAC            Name
lb1   Up   N/A       N/A   N/A N/A
NetIron#
```

Syntax: show interface brief wide

TABLE 23 Display of show interface ethernet port

This field...	Displays...
<*Module type*> <*Port#*> is <*State*>	The <*module type*> variable specifies a type of interface module, such as 10GigabitEthernet. The <*port#*> variable specifies the port number for the interface module. The <*state*> variable if the interface module is up or down.
Line protocol is <*status*>	The <*status*> variable specifies if the line protocol is up or down. If the interface is down due to Remote Fault, the reason is indicated as: "(remote fault)". If a port is down because of a Local Fault, the reason is indicated as: "(local fault)".
STP Root Guard is <*status*>	The <*status*> variable specifies if the STP Root Guard is enabled or disabled.
STP BPDU Guard is <*status*>	The <*status*> variable specifies if the STP BPDU Guard is enabled or disabled.
Hardware is <*module type*>	The <*module type*> variable specifies a type of interface module, such as <#>GigabitEthernet.
Address is <*MAC- address*>	The <*MAC- address*> variable specifies the MAC addess of the port.
Configured speed and actual speed	The speed that the module has been configured to operate at, and the actual speed it is currently operating at.
Configured port speed and actual duplex	The port capacity that the module has been configured to operate at, and the actual speed it is currently operating at.
Member of <*VLAN #*> (untagged) <*port#*> L2 VLANS (tagged) Port is in <dual mode/untagged/tagged> mode Port state is <*status*>	The <*VLAN#*> (untagged) variable specifies a port that is a member of only 1 VLAN. The <*port#*> L2 VLANS (tagged) variable specifies a port that is a member of multiple ports and untagged. A port is in <*dual- mode*> specifies member VLAN ports as untagged and tagged. The default mode is dual-mode. The <*status*> variable identifies the flow of traffic as forwarding or disabled.
STP configured to <*status*> Priority level Flow control <*status*>	The <*status*> variable specifies if the STP is ON or OFF. The priority level assigned to the port-based VLAN. The priority level is on scale from 0-7. The default is 0. The <*status*> variable is enabled or disabled.
Priority force <*status*>	The <*status*> variable specifies if the priority force on a port is disabled on enabled.
Drop precedence level <*value*>	Identifies the TOS or DSCP value in the IPv4 or IPv6 packet header. The <*value*> variable specifies the drop precedence on a scale from 0-3. Packets that contain a DSCP value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped. The default value is 0.
Drop precedence force <*status*>	The <*status*> variable specifies the drop precedence force as enabled or disabled. Identifies the drop precedence if the force command is configured for a specific ingress port.
arp-inspection-trust configured to <*status*>	The <*status*> variable specifies if arp-inspection-trust feature is configured ON or OFF. The default trust setting for a port is untrusted.
Mirror <*status*>	The <*status*> variable specifies if the port mirror command is configured as enabled or disabled.
Monitor <*status*>	The <*status*> variable specifies if the port monitor command is configured as enabled or disabled.

TABLE 23 Display of show interface ethernet port (Continued)

This field...	Displays...
<*Trunk membership*>	The <*Trunk membership*> variable identifies the interface module as a member of a primary or seconday port. This specifies members of an active port or not a member of an active port.
<*Configured trunk membership*>	The <*Configured trunk membership*> variable identifies the interface module as a member of any configured trunk or not a member of a configured trunk.
<*Port name*>	The <*port name*> variable identifies the name assigned to the port.
MTU <*# bytes*>, encapsulation ethernet	Maximum Transmision Unit (MTU) refers to the size of the largest packet or frame that a known layer can pass forward. The <*# bytes*> variable refers to size of the packet or frame.
<*#seconds*> input rate: <*value*> bits/sec, <*value*> packets/sec, <%> utilization	The <*#second*> input rate refers to: • The <*value*> of bits received per second. • The <*value*> of packets received per second. • The <%> utilization specifies the port's bandwidth used by received traffic.
<*# seconds*> output rate: <*value*> bits/sec, <*value*> packets/sec, <%> utilization	The <*#second*> output rate refers to: • The <*value*> of bits transmitted per second. • The <*value*> of packets transmitted per second. • The <%> utilization specifies the port's bandwidth used by transmitted traffic.
<*value*> packets input, <*value*> bytes, <*value*> no buffer	• The <*value*> variable specifies the number of packets received. • The <*value*> variable specifies the number of bytes received. • The <*value*> no buffer variable specifies the total number of packets that have been discarded by the MAC device, due to temporary inability to store the packets before forwarding to the Network Processor (NP).
Received <*value*> broadcasts, <*value*> multicasts, <*value*> unicasts	The <*value*> variable specifies the amount of traffic the interface module is receiving on broadcasts, multicasts, and unicast traffic.
<*value*> input errors, <*value*> CRC, <*value*> frame, <*value*> ignored	• The <*value*> input errors variable specifies the number of received packets with errors. • The <*value*> CRC variable specifies the number of packets discarded by the MAC device due to detected CRC error. • The <*value*> variable specifies the number of received packets with alignment errors. • The <*value*> variable specifies the number of received packets that are discarded. These parameters are not currently supported and will always display 0.
<*value*> runts, <*value*> giants	The <*value*> runts variable specifies the number of small packets that are less than 64 bytes. The <*value*> giants variable specifies the number of large packets greater than 1518 bytes. These parameters are not currently supported and will always display 0.
NP received	The number of packets received on the Network Processer (NP).
NP transmitted	The number of packets sent from the Network Processor to the Traffic Manager (TM).
NP Ingress dropped	The number of ingress packets dropped on the Network Processor.

TABLE 23 Display of show interface ethernet port (Continued)

This field...	Displays...
<*value*> packets output <*value*> bytes	• The <*value*> variable specifies the number of transmitted packets. • The <*value*> variable specifies the number of transmitted bytes.
Transmitted <*value*> broadcasts, <*value*> multicasts, <*value*> unicasts	The <*value*> variable specifies the amount of traffic the interface module transmitted on broadcasts, multicasts, and unicast traffic.
<*value*> output errors, <*value*> collisions	• The <*value*> variable specifies the number of transmitted packets with errors. • The <*value*> variable specifies the number of packets that experienced muli-access collisions. These parameters are not currently supported and will always display 0.
Network Processor transmitted <*value*> packets Received from Traffic Manager <*value*> packets	The <*value*> variable specifies the number of packets transmitted from the Network Processor. The <*value*> variable specifies the number of packets received by the Network Processor from the Traffic Manager.

Displaying statistics information for an Ethernet port

You can view statistical information about the traffic passing through a specified Ethernet port in one of two ways. The **monitor** commands allow you to monitor traffic statistics in real time, while the **show statistics** command provides a snapshot of the most recent traffic statistics.

Monitoring Ethernet port statistics in real time

You can monitor Ethernet traffic statistics in real time for a single port or traffic counters for all Ethernet ports using the **monitor** commands. When you execute a **monitor** command it retrieves and displays traffic statistics once per polling interval (2 seconds by default) until you pause or stop the display. The terminal window is fully occupied by the real-time display, and the command prompt is replaced by a footer listing options for pausing, canceling or modifying the display. When real-time monitoring is canceled, the command prompt is restored and the CLI resumes normal operation.

The following considerations affect the use of the **monitor** commands:

- Real-time monitor commands can be executed via Telnet, SSH, or a console session. Because of the slower communication rate in a console session, Dell recommends executing the **monitor** commands *only* from a Telnet or SSH session. The default poll interval for telnet and SSH is 2 seconds, but the default polling interval for a console session is 8 seconds. If you execute **monitor** commands from a console session, flickering of the display may occur.
- If the **monitor** command is executed in a console session, console debug messages will not be displayed on the console screen.
- When the **monitor** command is executed via telnet or SSH, debug messages will not be displayed during execution of the command even with a **debug destination telnet** <*session*> configuration present.

- **monitor** commands, in general, display two kinds of statistics: aggregated (counted since system startup or since last cleared using a **clear** command) and delta (counted since start of this **monitor** command or since last cleared using the **c** footer option on the monitor screen).
- Resizing of the terminal window is not supported during real-time statistics display. You must stop the execution of the command before resizing the terminal window.
- Terminal display size must be at least 80 characters wide by 24 lines in order to avoid garbled or truncated display.
- Execution of the **monitor** commands is unaffected by Telnet or SSH idle timeouts; as long as the **monitor** command is running, the terminal is not idle.
- There can be a noticeable impact on CPU utilization if the polling interval (monitor refresh interval) is short and multiple sessions are simultaneously executing **monitor** commands. When monitoring takes place by way of multiple simultaneous sessions, increase the polling interval to minimize impact on the CPU. (The polling interval/refresh rate ranges from 2 to 30 seconds, with a default value of 2 seconds for SSH or telnet connections and 8 seconds for a console session.)
- When you quit the **monitor** command, the CLI command prompt will usually be displayed at the bottom of the screen. If it appears instead in the middle of the screen, clear the screen using the command **cls** before executing further commands.

Real-time monitoring of traffic statistics for a specific Ethernet port

To monitor traffic statistics for a specific Ethernet port, enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# monitor statistics ethernet 1/2
```

Syntax: monitor statistics ethernet <*slot/port*>

The <*slot/port*> variable specifies the port for which you want to display statistics.

The **monitor statistics** command uses page mode display to show a detailed, port-specific traffic statistics screen which is updated every poll interval. You can modify the display using the commands shown in the footer. (Note that when you enter footer options they are not executed immediately but will be interpreted by the monitor engine during the next polling cycle.) The footer commands and their effects are described in Table 24.

TABLE 24 Footer commands for **monitor statistics** display

t	Displays the transmit/output statistics (the default) and continues the execution of the original command.
r	Displays the receive/input statistics and continues the execution of the original command.
n	Continues the execution of the command for the next available Ethernet interface. If there is no next interface available, the monitor continues to display statistics for the current interface.
p	Continues the execution of the command for the previous Ethernet interface. If there is no next interface available, the monitor continues to display statistics for the current interface.
c	Clears the current delta counters and continues the execution of the original command. To clear the aggregate counters, use the appropriate **clear** command.
f	Freezes the execution of the command; pauses retrieval and display of the statistics. While display is frozen, the only valid commands are **s** and **q**: you can restart or quit the monitor, but any other command will be ignored.

TABLE 24 Footer commands for **monitor statistics** display

s	Restarts the execution of the command; resumes retrieval and display of the statistics.
F	Decreases the polling interval (monitor refresh interval) by one second and continues the execution of the original command with the new refresh interval. This option will decrease the refresh interval until it is equal to 2 seconds, the minimum supported refresh interval value. The default value is 2 seconds. This command is not displayed in the footer of the statistics screen.
S	Increases the polling interval (monitor refresh interval) by one second and continues the execution of the original command with the new refresh interval. This option will increase the refresh interval until it is equal to 30 seconds, the maximum supported refresh interval value. The default value is 2 seconds. This command is not displayed in the footer of the statistics screen.
q or **escape** or **^c**	Quits the execution of the command and returns to the command prompt.
u	**MLX only**: Displays the first page of the multipage display (page-up operation).
d	**MLX only**: Displays the second page of the multipage display (page-down operation).

NetIron MLX Series example

```
NetIron# monitor statistics ethernet 4/1

                            Seconds: 8         poll: 8  Time: Aug 19 16:10:59

Page 1 of 2  Interface Tx Statistics              Current                Delta
Ethernet 4/1 Tx interface statistics
Traffic statistics:
 Out Packets                                   17083660926               533508
 Out Octets                                  1093354299264             34144512
 Out Unicast Packets                           17083660926               533508
 Out Multicast Packets                                 0                    0
 Out Broadcast Packets                                 0                    0

Error statistics:
 Out Errors                                            0                    0
 Out Discards                                          0                    0




Tx/Rx=t/r, Page1/2=u/d, Next/Prev=n/p, Clear=c :Freeze=f/s Quit=q
```

```
                          Seconds: 40        poll: 8  Time: Aug 19 16:11:31

Page 2 of 2   NP       Tx Statistics              Current                Delta
Ethernet 4/1 Tx NP statistics
 Sent to MAC Packet                          17085805774              2670758
 Raw Good Packet                             17085805774              2670758
 IPX HW Forwarded Packet                               0                    0
 Receive from TM                             17085805775              2670759
 Unicast Packet                              17085805774              2670758
 Broadcast Packet                                      0                    0
 Multicast Packet                                      0                    0
Error statistics :
 Bad Packet Count                                      0                    0

 ACL Drop                                              0                    0
 Source Port Supress Drop                              0                    0
 IPv4 Packet                                           0                    0
 IPv6 Packet                                           0                    0
 IPv4 Byte                                             0                    0
 IPv6 Byte                                             0                    0


Tx/Rx=t/r, Page1/2=u/d, Next/Prev=n/p, Clear=c :Freeze=f/s Quit=q
```

The previous output shows the first and second pages of the detailed traffic statistics display for Ethernet port 4/1 from a PowerConnect B-MLXe, displaying transmit counters (the default).

Real-time monitoring of traffic statistics for all Ethernet ports

To monitor summary traffic data (total packets or bytes sent and received) for all Ethernet ports (displaying up to 16 ports per screen), enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# monitor interface traffic

                          Seconds: 248          Time: Mar 11 20:12:08
Interface traffic statistics:
                 InPackets           Delta           OutPackets           Delta
  e1/1               24615            4004                24308            3986
  e1/2                   0               0                    0               0
  e1/3                   0               0                    0               0
  e1/4                   0               0                    0               0
  e1/5                   0               0                    0               0
  e1/6                   0               0                    0               0
  e1/7                   0               0                    0               0
  e1/8                   0               0                    1               1
  e1/9                   0               0                    0               0
 e1/10                   0               0                    0               0
 e1/11                   0               0                    0               0
 e1/12                   0               0                    0               0
 e1/13                   0               0                    0               0
 e1/14                   0               0                    0               0
 e1/15                   0               0                    0               0
 e1/16                   0               0                    0               0


Packets=p or Bytes=b, Delta=d or Rate=r, Clear=c, Next=n :Freeze=f/s Quit=q
```

Syntax: **monitor interface traffic** [**ethernet** <*slot/port*>]

The **monitor interface traffic** command uses page mode display to produce an updating statistics screen which is updated every poll interval and which can be modified using the commands shown at the bottom of the display. (Note that when you enter footer options they are not executed immediately but will be interpreted by the monitor engine during the next polling cycle.) Normally the display begins with the lowest numbered Ethernet port; the **ethernet** <*slot/port*> option starts the display instead with the specified port.

The footer commands and their effects are described inTable 25.

TABLE 25 Footer commands for **monitor interface traffic** display

p	Displays input/output packets instead of bytes and continues the execution of the original command.
b	Displays input/output bytes instead of packets and continues the execution of the original command.
d	Displays delta counters instead of rate counters and continues the execution of the original command.
r	Displays rate counters instead of delta counters and continues the execution of the original command.
c	Clears the current delta counters and continues the execution of the original command. To clear the aggregate counters, use the appropriate **clear** command.
n	Moves to the next group of interfaces and continues the execution of the original command.
f	Freezes the execution of the command; pauses retrieval and display of the statistics. While display is frozen, the only valid commands are **s** and **q**: you can restart or quit the monitor, but any other command will be ignored.
s	Restarts the execution of the command; resumes retrieval and display of the statistics.
F	Decreases the polling interval (monitor refresh interval) by one second and continues the execution of the original command with the new refresh interval. This option will decrease the refresh interval until it is equal to 2 seconds, the minimum supported refresh interval value. The default value is 2 seconds. This command is not displayed in the footer of the statistics screen.
S	Increases the polling interval (monitor refresh interval) by one second and continues the execution of the original command with the new refresh interval. This option will increase the refresh interval until it is equal to 30 seconds, the maximum supported refresh interval value. The default value is 2 seconds. This command is not displayed in the footer of the statistics screen.
q or **escape** or **^c**	Quits the execution of the command and returns to the command prompt.

Displaying recent traffic statistics for an Ethernet port

To display information from the **show statistics** command for an Ethernet port, enter the following command at any CLI level.

```
NetIron# show statistics ethernet 9/1
PORT 9/1 Counters:
          InOctets       210753498112           OutOctets      210753550720
             InPkts        1646511726              OutPkts        1646512119
     InBroadcastPkts                 0     OutBroadcastPkts                 0
     InMulticastPkts                 0    OutMulticastPkts                 0
       InUnicastPkts        1646511726      OutUnicastPkts        1646512142
          InDiscards                 0         OutDiscards                 0
           InErrors                  0          OutErrors                  0
        InCollisions                 0       OutCollisions                 0
                                         OutLateCollisions                 0
           Alignment                 0                 FCS                 0
           GiantPkts                 0           ShortPkts                 0
         InBitsPerSec        3440829770       OutBitsPerSec        3440686411
         InPktsPerSec           3360185        OutPktsPerSec           3360085
        InUtilization             39.78%     OutUtilization             39.78%
```

Syntax: show statistics ethernet <*slot/port*>

The <*slot/port*> variable specifies the port that you want to display statistics for.

This field...	Displays...
InOctets	The total number of good octets and bad octets received.
OutOctets	The total number of good octets and bad octets transmitted.
InPkts	The total number of packets received. The count includes rejected and local packets that are not transmitted to the switching core for transmission.
OutPkts	The number of good packets received. The count includes unicast, mulicast, and broadcasts packets.
InBroadcastPkts	The total number of good broadcast packets received.
OutBroadcastPkts	The total number of good braodcast packets transmitted.
InMulticastPkts	The total number of good multicast packets received.
OutMulticastPkts	The total number of good multicast packets transmitted.
InUnicastPkts	The total number of good unicast packets received.
OutUnicastPkts	The total number of good unicast packets transmitted.
InDiscards	The total number of packets that were received and then dropped due to a lack of receive buffers.
OutDiscards	The total number of packets that were transmitted and then dropped due to a lack of transmit buffers.
InErrors	The total number of packets received that had Alignment errors or phy errors.
OutErrors	The total number of packets transmitted that had Alignment errors or phy errors.
InCollisions	The total number of packets received in which a Collision event was detected.
OutCollisions	The total number of packets transmitted in which a Collision event was detected.
OutLateCollisions	The total number of packets transmitted in which a Collision event was detected, but for which a receive error (Rx Error) event was not detected.
Alignment	The total number of packets received that were from 64 – 1518 octets long, but had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error).

This field...	Displays...
FCS	The Frame Checksum error.
GiantPkts	The total number of packets for which all of the following was true: • The data length was longer than the maximum allowable frame size. • No Rx Error was detected. This counter is only for 10GbE interfaces.
ShortPkts	The total number of packets received for which all of the following was true: • The data length was less than 64 bytes. • No Rx Error was detected. • No Collision or late Collision was detected.
InBitsPerSec	The number of bits received per second.
OutBitsPerSec	The number of bits transmitted per second.
InPktsPerSec	The number of packets received per second.
OutPktsPerSec	The number of packets transmitted per second.
InUtilization	The percentage of the port's bandwidth used by received traffic.
OutUtilization	The percentage of the port's bandwidth used by transmitted traffic.

Configuring SNMP to revert ifType to legacy values

The ifType for all Ethernet interfaces (10/100/1G/10G) returns the value ethernetCsmacd(6) as mandated by RFC 2665. If you want ifType to return gigabitEthernet (117) or fastEther(62) for Ethernet interfaces, enter the following command.

```
NetIron(config)# snmp-server legacy iftype
```

Syntax: [no] snmp-server legacy iftype

When this command is configured, the values gigabitEthernet (117) or fastEther(62) are returned for ifType. If you issue a **no snmp-server legacy iftype,** ifType returns ethernetCsmacd(6) for Ethernet interfaces.

Configuring snAgentConfigModuleType to return original values

Enumeration values for snAgentConfigModuleType object in the SNMP MIB have been changed for the Product Name to resolve enumeration conflicts with other hardware modules in the IronWare enterprise MIB. For example, an SNMP get of the snAgentConfigModuleType of the 10x1GC module returned mlxe20PortGigCopperSPModule(84). The snAgentConfigModuleType returns fdryMlxe20PortGigCopperSPModule(1084) for the 10x1GC module.

If you want snAgentConfigModuleType to return the enumeration values, configure the following command.

```
NetIron(config)# snmp-server legacy module-type
```

Syntax: [no] snmp-server legacy module-type

Refer to the *IronWare MIB Reference* for details on snAgentConfigModuleType.

Preserving interface statistics in SNMP

By default, statistics for an interface is cleared from both the CLI and SNMP when the following commands are entered on the CLI:

- **clear statistics ethernet** *slot-number/port-number*
- **clear statistics pos** *slot-number/port-number*
- **clear rmon statistics**
- **clear statistics log** *slot-number/port-number*

If you want to preserve interface statistics in SNMP when these commands are entered, configure the following command at the Global level of the CLI.

```
NetIron(config)# snmp-server preserve-statistics
```

Syntax: [no] snmp-server preserve-statistics

For details on which interface statistics are preserved in SNMP, refer to the "Preserved interface statistics for SNMP" section of the "Supported Standard MIBs" chapter in the *IronWare MIB Reference*.

NOTE
Statistics for an interface will be different between the CLI and SNMP if **snmp-server preserve-statistics** is configured and the clear commands listed above are executed.

Configuring Interface Parameters 4

Overview

The following Interface Parameters features are supported by the NetIron MLX Series devices.

- Assigning Port Name
- Assigning an IP Address to a Port
- Modifying Port Speeds
- Default Gigabit Negotiation Mode
- Flow Control
- Port Transition Hold Timer
- Port Flap Dampening
- Mirror Port and Monitor Ports
- ACL-based Inbound Mirroring
- Setting IP VPN Packets
- 10G WAN PHY Fault and Performance Management
- 10G Interface Numbering
- Wait for all Cards
- CAM Sharing
- Link Fault Signaling
- 10G Port Local Fault
- LFS or RFN Counters for 10G LAN PHY
- Displaying Network Processor Statistics
- Port loop detection
- Auto negotiation speed limit

NOTE
To modify Layer 2, Layer 3, or Layer 4 features on a port, refer to the appropriate section in this chapter or other chapters. For example, to modify Spanning Tree Protocol (STP) parameters for a port, refer to "Changing STP port parameters" on page 374.

All PowerConnect ports are pre-configured with default values that allow the device to be fully operational at initial startup without any additional configuration. However, in some cases, changes to the port parameters may be necessary to adjust to attached devices or other network requirements.

Assigning a port name

A port name can be assigned to help identify interfaces on the network. You can assign a port name to physical ports, virtual routing interfaces, and loopback interfaces.

To assign a name to a port, enter the following command.

```
NetIron(config)# interface e 2/8
NetIron(config-if-e10000-2/8)# port-name Marsha Markey
```

Syntax: [no] port-name <*text*>

The <*text*> parameter is an alphanumeric string. The name can have up to 255 characters on a PowerConnect and can include blanks. You do not need to use quotation marks around the string, even when it contains blanks.

Assigning an IP address to a port

To assign an IP address to an interface, enter the following commands.

```
NetIron(config)# interface e 1/8
NetIron(config)# ip address 192.45.6.110 255.255.255.0
```

Syntax: [no] ip address <*ip-addr*> <*ip-mask*>

or

Syntax: [no] ip address <*ip-addr*>/<*mask-bits*>

NOTE
You also can enter the IP address and mask in CIDR format, as follows.

```
NetIron(config)# ip address 192.45.6.1/24
```

Modifying port speed

Each of the 10/100/1000BaseTX ports is designed to auto-sense and auto-negotiate the speed and mode of the connected device. If the attached device does not support this operation, you can manually enter the port speed to operate at either 10 Mbps or 100 Mbps. The default value is 10 or 100 half- or full-duplex.

NOTE
Modifying the port speed of a port that has a pre-configured rate limit policy may result in the inability to remove the port's rate limit policy.

To change the port speed of interface 1/8 from the default of 10/100 auto-sense to 10 Mbps operating at full-duplex, first disable the port. Then, enter the following.

```
NetIron(config)# interface e 1/8
NetIron(config-if-e10000-1/8)# speed-duplex 10-full
```

Syntax: [no] speed-duplex <*value*>

The <*value*> can be one of the following:

- 10-full
- 10-half
- 100-full
- 100-half
- 1000-full
- 1000-half
- auto

The default is auto.

NOTE
An auto negotiation port must be connected to another auto negotiation port. If you connect an auto negotiation port to a fixed speed or duplex port, the behavior is undefined.

Also, ports must be disabled before changing speed.

Modifying port mode

You can configure a port to accept either full-duplex (bi-directional) or half-duplex (uni-directional) traffic. Port duplex mode and port speed are modified by the same command.

To change the port speed of interface 1/8 from the default of 10/100 auto-sense to 10 Mbps operating at full-duplex, enter the following command.

```
NetIron(config)# interface e 1/8
NetIron(config-if-e10000-1/8)# speed-duplex 10-full
```

Syntax: speed-duplex <*value*>

The <*value*> can be one of the following:

- 10-full
- 10-half
- 100-full
- 100-half
- 1000-full
- 1000-half
- auto

The default is auto.

Auto Negotiation Speed Limit

Auto-negotiation is an active method of determining the link mode. Each interface is expected to transmit specific information in a specific format. If an interface that is expecting to use auto-negotiation does not receive this information from the other side, it assumes the other side cannot detect or change its mode.

One of the most common causes of performance issues on 10/100/1000 Mb Ethernet links occurs when one port on the link operates at half-duplex while the other port operates at full-duplex. This occurs when one or both ports on a link are reset and the auto-negotiation process does not result in both link partners having the same configuration. It also can occur when users reconfigure one side of a link and forgets to reconfigure the other side. Both sides of a link should have auto-negotiation on, or both sides should have it off.

The auto negotiation speed limit feature allows the user to reduce or limit the port speed when auto- negoiation is configured. You can set the port to automatically reduce the speed from 1000Mb to 100Mb or 10Mb. The **down-shift** option will reduce the port speed to 100Mb from 1000Mb automatically once a 2-wire cable is detected.

The auto negotiation speed limit feature is supported only on FIXED (non SFP) copper ports when auto-neg is ON.

```
NetIron(config)# interface e 1/8
NetIron(config-if-e10000-1/8)# auto
NetIron(config-if-e10000-1/8)# link-config gig copper autoneg-control down-shift
```

Syntax: [no] link-config gig copper autoneg-control [down-shift|100m|10m]

The **10m** option will limit the port to negotiation to speeds and duplex of 10mb.

The **100m** option will limit the port to negotiation of speeds and duplex below 100mb.

Disabling or re-enabling a port

The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. he default value for a port is disabled.

To disable port 8 on module 1 of a PowerConnect router, enter the following command.

```
NetIron(config)# interface e 1/8
NetIron(config-if-e10000-1/8)# disable
```

Syntax: [no] disable

Syntax: [no] enable

You also can disable or re-enable a virtual routing interface. To do so, enter commands such as the following.

```
NetIron(config)# interface ve v1
NetIron(config-vif-1)# disable
```

Syntax: [no] disable

To re-enable a virtual routing interface, enter the **enable** command at the Interface configuration level. For example, to re-enable virtual routing interface v1, enter the following command.

```
NetIron(config-vif-1)# enable
```

Syntax: [no] enable

Disabling Source Address Learning on a port

The default operation is for Source Address (SA) Learning to be enabled on all ports. It can be useful to disable SA Learning on a port in situations where high CPU usage is occuring because a large number of packets are being sent to the CPU for SA Learning. For example, it can be useful to disable SA Learning on physical ports that are part of a Virtual Ethernet (VE) interface that has no need to switch packets.

SA Learning can be disabled on a port using the **sa-learning-disable** command as shown in the following.

```
NetIron(config)# interface e 1/8
NetIron(config-if-e10000-1/8)# sa-learning-disable
```

Syntax: [no] sa-learning-disable

Changing the default Gigabit negotiation mode

You can configure the default Gigabit negotiation mode to be one of the following:

- **neg-full-auto** – The port first tries to perform a negotiation its peer port to exchange capability information. If the other port does not respond, the port reverts to the Negotiation-off state.
- **auto-gig** – The port tries to performs a negotiation with its peer port to exchange capability information. This is the default state.
- **neg-off** – The port does not try to perform a negotiation with its peer port.

Unless the ports at both ends of a Gigabit Ethernet link use the same mode (either **auto-gig** or **neg-off**), the ports cannot establish a link. An administrator must intervene to manually configure one or both sides of the link to enable the ports to establish the link.

Changing the negotiation mode

You can change the negotiation mode for individual ports as shown in the following.

```
NetIron(config)# interface ethernet 4/1 to 4/4
NetIron(config-mif-4/1-4/4)# gig-default neg-off
```

This command changes the default **auto-gig** setting and sets the negotiation mode to neg-off for ports 4/1 – 4/4.

Syntax: [no] gig-default neg-full-auto | auto-gig | neg-off

The **neg-full-auto, auto-gig**, and **neg-off** options are as described above.

Disabling or re-enabling flow control

You can configure full-duplex ports on a system to operate with or without flow control (802.3x).

NOTE
The **flow-control** command has been deprecated, and will no longer be accepted except at system bootup time.

The new command is **system global-flow-control** command. The **system global-flow-control** command is enabled by default.

To disable flow control on full-duplex ports on a system, enter the following command.

```
NetIron(config)# no system global-flow-control
```

To turn the feature back on, enter the following command.

```
NetIron(config)# system global-flow-control
```

Syntax: [no] system global-flow control

Specifying threshold values for flow control

The 802.3x flow control specification provides a method for slowing traffic from a sender when a port is receiving more traffic than it can handle. Specifically, the receiving device can send out 802.3x PAUSE frames that request that the sender stop sending traffic for a period of time.

The PowerConnect router generates 802.3x PAUSE frames when the number of buffers available to a module's Buffer Manager (BM) drops below a threshold value. A module's BM can start running out of buffers when a port receives more traffic than it can handle. In addition, the device drops the lowest priority traffic when the number of available buffers drops below a second threshold. When the number of available buffers returns to a higher level, the device sends out another PAUSE frame that tells the sender to resume sending traffic normally. You can specify values for both thresholds, as well as the module where the thresholds are to take effect.

NOTE
sflow is implemented in the default VRF only. Therefore, sflow data is only accessible by the sflow collector (sflow destination host(s)) defined in the default VRF.

NOTE
To use this feature, 802.3x flow control must be enabled globally on the device. By default, 802.3x flow control is enabled on the PowerConnect router, but can be disabled with the **no flow-control** command.

To specify threshold values for flow control, enter the following command.

```
NetIron(config)# qd-flow sink 75 sunk 50 slot 1
```

Syntax: [no] qd-flow sink <*sinking-threshold*> **sunk** <*sunk-threshold*> **slot** <*slot*>

The threshold values are percentages of the total number of buffers available to a module's Buffer Manager.

When the <*sinking-threshold*> is reached, the PowerConnect router sends out 802.3x PAUSE frames telling the sender to stop sending traffic for a period of time.

When the <*sunk-threshold*> is reached, the PowerConnect router drops traffic at the specified priority level.

The <*slot*> parameter specifies the location of the module where the thresholds are to take effect.

Modifying port priority (QoS)

You can give preference to the inbound traffic on specific ports by changing the Quality of Service (QoS) level on those ports. For information and procedures, refer to 9, “Configuring Quality of Service for the NetIron MLX”.

Setting IP VPN packets with a TTL value of 1 to be dropped

This command is for IP VPN packets only. Under normal conditions IP VPN packets with a TTL value equal to 0 are always dropped in hardware regardless of the setting of this command. With this command set, IP VPN packets with TTL value equal to one will also be dropped in hardware.

To enable this command use the following command.

```
NetIron(config)# interface ethernet 4/1 to 4/4
NetIron(config-if-4/1)# hw-drop-bad-ttl-pkt
```

Syntax: [no] hw-drop-bad-ttl-pkt

The default value is off.

Port transition hold timer

Using the **delay-link-event** command will delay the sending of port "up" or "down" events to Layer 2 protocols. While link down events are reported immediately in syslog, their effect on higher level protocols such as OSPF is delayed according to how the delay-link-event is configured. This command affects the physical link events. However, the resulting logical link events are also delayed. This is a per-interface command.

For example, if VSRP is enabled on the port, the ownership will not change until the port status has remained up or down for the configured amount of time to ensure that minor transient states of a link do not unintentionally cause a disruptive topology change in the network.

NOTE
All LAG ports must have the same delayed-link-down-event configuration.

The following command will delay the sending of port "down" event for 100ms when a port state is detected "down". If the port state is detected "up" afterwards within 100ms, the delayed "down" event is cancelled; otherwise, the "down" event is sent after 100ms. This allows the upper layer applications not to be affected by a port state flapping.

```
NetIron (config-if-e1000-1/2)# delay-link-event 2 down
```

Syntax: [no] delay-link-event <*time*> **up | down**

The <*time*> parameter is the number of 50-ms units. The default is 0. The valid range is from 0 to 200.

The **up** parameter means only "up" events are delayed.

The **down** parameter means that only the down events are delayed.

If neither the **up** or **down** parameter is specified, both up and down events are delayed. This is the default.

Port flap dampening

The port flap dampening feature allows you to configure a wait period before a port, whose link goes down then up, becomes enabled.

If the port link state toggles (from down to up or from up to down) for a specified number of times within a specified period, the interface is physically disabled for the specified wait period. Once the wait period expires, the port's link state is re-enabled. However, if the wait period is set to zero (0) seconds, or you want to re-enable the port before the wait period expires, the port must be manually re-enabled as described in "Re-enabling a port disabled by port link dampening" on page 140.

Configuring port link dampening on an interface

This feature is configured at the interface level.

```
NetIron(config)#interface ethernet 2/1
NetIron(config-if-e10000-2/1)#link-error-disable 10 3 10
```

Syntax: [no] link-error-disable <*toggle-threshold*> <*sampling-time-in-sec*> <*wait-time-in-sec*>

The <*toggle-threshold*> is the number of times a port's link state goes from up to down and down to up before the wait period is activated. The default is 0. Enter a valid value range from 1-50.

The <*sampling-time-in-sec*> is the amount of time during which the specified toggle threshold can occur before the wait period is activated. The default is 0 seconds. Enter a value between 1 and 65565 seconds.

The <*wait-time-in-sec*> is the amount of time the port remains disabled (down) before it becomes enabled. Entering 0 indicates that the port will stay down until an administrative override occurs. Enter a value between 0 and 65565 seconds.

Configuring port link dampening on a LAG

You can configure the port link dampening feature on the primary port of a LAG at the interface level using the **link-error-disable** command. Once configured on the primary port of the LAG, the feature is enabled on all port that are members of the LAG. You cannot disable the feature from a member of the LAG.

Enter commands such as the following on the primary port of a LAG.

```
NetIron(config)#interface ethernet 2/1
NetIron(config-if-e10000-2/1)#link-error-disable 10 3 10
```

Re-enabling a port disabled by port link dampening

A port disabled by the port link dampening is automatically re-enabled once the wait period expires; however, if the wait period is set to zero (0) seconds or you want to re-enable the port before the configured wait period expires, you must re-enable the port by entering the **link-error-disable** command on the disabled port as shown in the following.

```
NetIron(config)#interface ethernet 2/1
NetIron(config-if-e10000-2/1)#link-error-disable 10 3 10
```

NOTE
You must enter the **link-error-disable** command with the <*toggle-threshold*> <*sampling-time-in-sec*> and <*wait-time-in-sec*> variables defined to re-enable the port. Using the **link-error-disable** command without the variables, will not bring the port back up.

Displaying ports configured with port link dampening

Ports that have been disabled due to the port link dampening feature are not identified in a **show running-config** command.

Use the **show interface link-error-disable** to display the ports that have the port link dampening feature enabled.

```
NetIron(config-if-e10000-8/1)#show interfaces link-error-disable
Port 8/1: link-error-disabled (Config: 2 toggles per 3 sec, wait time 1 sec)
Port 8/3: not link-error-disabled (Config: 2 toggles per 2 sec, wait time 30 sec)
Port 8/4: not link-error-disabled (Config: 2 toggles per 2 sec, wait time 30 sec)
```

TABLE 26 link-error-disable

Displays...	Description...
port	The port that has been configured
link-error-disabled	The port that has been disabled by this feature
not link-error-disabled	The “not” means the port has not been disabled due to this feature
toggle	The number of times a port’s link state goes from up to down and down to up before the wait period is activated
wait time	The amount of time the port remains disabled (down) before it becomes enabled

Issuing the **disabled-only** with the command displays only the ports that have been disabled by the port link dampening feature.

```
NetIron(config-if-e10000-8/1)#show interfaces link-error-disable disabled-only
Port 8/1: link-error-disabled (Config: 2 toggles per 3 sec, wait time 1 sec)
```

Syntax: show interface link-error-disable [disabled-only]

Entering the **show interface link-error-disable** displays all the ports that have the port link dampening feature enabled. Add the **disabled-only** keyword for a list of ports disabled by this feature.

Port loop detection

This feature allows the Dell device to disable a port that is on the receiving end of a loop by sending test packets. You can configure the time period during which test packets are sent.

Strict mode and Loose mode

There are two types of loop detection; Strict Mode and Loose Mode. In Strict Mode, a port is disabled only if a packet is looped back to that same port. Strict Mode overcomes specific hardware issues where packets are echoed back to the input port. In Strict Mode, loop detection must be configured on the physical port.

In Loose Mode, loop detection is configured on the VLAN of the receiving port. Loose Mode disables the receiving port if packets originate from any port or VLAN on the same device. The VLAN of the receiving port must be configured for loop detection in order to disable the port.

Recovering disabled ports

Once a loop is detected on a port, it is placed in a disabled state. The port will remain disabled until one of the following occurs:

- You manually disable and enable the port at the Interface Level of the CLI
- You enter the command **clear loop-detection**. The **clear loop-detection** command clears the loop detection statistics and enables all disabled ports
- The device automatically re-enables the port. To set your device to automatically re-enable disabled ports, refer to "Configuring the device to automatically re-enable ports" on page 144.

Disable duration and loop detection interval

By default, the ports are shutdown permanently until user enables it manually. You can configure the disable duration from 1 minute to 1440 minutes (24 hours)

By default. the Loop Detection time Interval between the loop detection BPDU is 1 second. You can configure the loop detection PDU interval from 100ms to 10 seconds.

Configuration notes

Loopback detection packets are sent and received on both tagged and untagged ports. Therefore, this feature cannot be used to detect a loop across separate devices.

The following information applies to Loose Mode loop detection:

- Loop detection is configured on the VLAN. Different VLANs may disable different ports. A disabled port affects every VLAN using it.
- Loose Mode disables the receiving port if packets originate from any port or member port of a VLAN on the same device
- The VLAN of the receiving port must be configured for loop detection in order to disable the port.
- Loose Mode floods test packets to the entire VLAN. This can impact system performance if too many VLANs are configured for Loose Mode loop detection.

The following information applies to Strict Mode loop detection:

- A port is disabled only if a packet is looped back to that same port.
- Loop detection must be configured on the physical port.
- Strict Mode overcomes specific hardware issues where packets are echoed back to the input port.

NOTE
Dell recommends that you limit the use of Loose Mode. If you have a large number of VLANS or VLAN groups, configuring loop detection on all of them can significantly affect system performance because of the flooding of test packets to all configured VLANs. An alternative to configuring loop detection in a VLAN-group of many VLANs is to configure a separate VLAN with the same tagged port and configuration, and enable loop detection on this VLAN only.

NOTE
When loop detection is used with L2 loop prevention protocols, such as spanning tree (STP), the L2 protocol takes higher priority. Loop detection cannot send or receive probe packets if ports are blocked by L2 protocols, so it does not detect L2 loops when STP is running because loops within a VLAN have been prevented by STP. Loop detection running in Loose Mode can detect and break L3 loops because STP cannot prevent loops across different VLANs. In these instances, the ports are not blocked and loop detection is able to send out probe packets in one VLAN and receive packets in another VLAN. In this way, loop detection running in Loose Mode disables both ingress and egress ports.

Enabling loop detection

Use the loop-detection command to enable loop detection on a physical port (Strict Mode) or a VLAN (Loose Mode). Loop detection is disabled by default. The following example shows a Strict Mode configuration.

```
NetIron(config)#interface ethernet 1/1
NetIron(config-if-e1000-1/1)#loop-detection
```

The following example shows a Loose Mode configuration.

```
NetIron(config)#vlan 20
NetIron(config-vlan-20)#loop-detection
```

The following example shows a Loose Mode configuration for a VLAN group.

```
NetIron(config)#vlan-group 10
NetIron(config-vlan-group-10)#add-vlan 1 to 100
NetIron(config-vlan-group-10)#loop-detection
```

By default, the port will send test packets every one second, or the number of seconds specified by the **loop-detection-interval** command. Refer to "Configuring a global loop detection interval."

Syntax: [**no**] **loop-detection**

Use the [**no**] form of the command to disable loop detection.

Configuring a global loop detection interval

The loop detection interval specifies how often a test packet is sent on a port. When loop detection is enabled, the loop detection time unit is 0.1 second, with a default of 10 (one second). The range is from 1 (one tenth of a second) to 100 (10 seconds). You can use the show loop-detection status command to view the loop detection interval.

To configure the global loop detection interval, enter a command such as the following.

```
NetIron(config)#loop-detection-interval 50
```

This command sets the loop-detection interval to 5 seconds (50 x 100ms).

To revert to the default global loop detection interval of 10, enter one of the following.

```
NetIron(config)#loop-detection-interval 10
```
OR
```
NetIron(config)#no loop-detection-interval 50
```

Syntax: [**no**] **loop-detection-interval** <*number*>

Where <*number*> is a value from 1 to 100. The system multiplies your entry by 0.1 to calculate the interval at which test packets will be sent.

Configuring the device to automatically re-enable ports

To configure the Dell device to automatically re-enable ports that were disabled because of a loop detection, enter the following command The default is 0.

```
NetIron(config)#loop-detection disable-duration 1440
```

The above command will cause the Dell device to automatically re-enable ports that were disabled for a duration of 24 hours because of a loop detection. This configuration applies to all the ports that are configured the loop detection (strict or loose).

Syntax: [**no**] **loop-detection disable-duration** <num>

Use the [**no**] form of the command to disable this feature.

Where <*num*> is the number of minutes from 0 to 1440. When 0 is specified, it is permanently off.

Clearing loop-detection

To clear loop detection statistics and re-enable all ports that are in disabled state because of a loop detection, enter the following command.

```
NetIron #clear loop-detection
```

Syntax: **clear loop-detection** [vlan | ethernet] <vlanid/port-num>

Where **port-num** enables the specified port.

Where **vlan-id** enables all the ports disabled by loop detection for this VLAN

Displaying loop-detection information

Use the **show loop-detection** command to display the loop detection status.

```
NetIron(config-vlan-100)#show loop-detection

loop detection packets interval: 10 (unit 100 msec)
loop detection disable duration: 10 (In minutes, 0 means permanently disabled)

Ports mode loop detection
=========================
port-num   disable-count

1/12       0
1/11       0

Vlan mode loop detection
========================
vlan-id    disable-count

100        2
10          0
200        0

Ports disabled by loop detection
================================
port       age(minutes) disable cause

 1/11  1            Disabled by VLAN: 100 loopdetect 1/11
 1/12  1            Disabled by VLAN: 100 loopdetect 1/12
```

Syntax: show loop-detection

TABLE 27 Port loop detection output description

Parameter	Description
loop detection packets interval	Specifies how often a test packet is sent on a port.
loop detection disable duration	Specifies the device to automatically re-enable ports that were disabled for the configured duration because of a loop detection
ports mode	The VLAN or port that port loop detection was configured on.
loop detection disabled ports	The ports that are disabled by port loop detection. • port - The port number that was disabled by port loop detection. • age - The time duration after which port will be automatically re-enabled. If the age is “0”, it means port is not configured to be automatically re-enabled. • disable cause - specifies all the ports that were disabled by loop detection (either strict or loose).

Syslog message

The following message is logged when a port is disabled due to loop detection. This message will also appear on the console.

```
SYSLOG: Jan 27 18:16:42:<14>Jan 27 18:16:42 LOOP_DETECT LOG: Port Down 1/10 -
Loop detected on VLAN: 150
```

Mirroring and Monitoring

You can monitor traffic on PowerConnect router ports by configuring another port to “mirror” the traffic on the ports you want to monitor. By attaching a protocol analyzer to the mirror port, you can observe the traffic on the monitored ports.

Monitoring traffic on a port is a two-step process:

- Enable a port to act as the mirror port. This is the port to which you connect your protocol analyzer.
- Enable monitoring on the ports you want to monitor.

You can monitor input traffic, output traffic, or both.

Any port on a module can operate as a mirror port and you can configure more than one mirror port. You can configure the mirror ports on different modules and you can configure more than one mirror port on the same module.

Configuration guidelines for monitoring traffic

Use the following considerations when configuring mirroring for inbound and outbound traffic:

- Any port can be mirrored and monitored except for the management port.
- Only one inbound mirror port can be configured for any inbound monitor port.
- Only one outbound mirror port can be configured for any outbound monitor port.
- A LAG port can be configured as either an inbound or outbound monitor port.
- A LAG port cannot be configured as either an inbound or an outbound mirror port.
- Both input and output monitoring are supported.
- Monitoring for LAG ports is supported.
- sFlow and monitoring can be enabled concurrently on the same port.
- ACL-based inbound mirroring is supported.
- ACL-based inbound sFlow is not concurrently supported.

Assigning a mirror port and monitor ports

To configure ethernet port 3/1 for port mirroring, enter the following command.

```
NetIron(config)# mirror-port ethernet 3/1
```

Syntax: [no] **mirror-port ethernet** <*slot*>/<*portnum*> | **pos** <*slot*>/<*portnum*>

POS ports can be monitored. However, because a POS port cannot be configured as a monitor port using the **monitor ethernet** command described below, the monitored POS port must be mirrored through an Ethernet port.

NOTE
If a port is configured as a mirror port, all traffic sent from that port will retain the encapsulation of the port being monitored and not add the encapsulation of the Egress port.

Enter the slot and port number of the port that will be the mirrored.

```
NetIron(config)# interface ethernet 4/1
NetIron(config-if-4/1)# monitor ethernet 3/1
```

Syntax: [no] monitor ethernet <*slot*>/<*portnum*> **both | input | output**

Enter the slot and port number of the port that will serve as the monitor port. This port cannot be the same as the mirror port.

NOTE
A mirror port must be an Ethernet port. While a a POS port can be monitored, a POS port cannot be used as a mirror port. Also, POS control packets and IS-IS control packets transmitted or received at a POS port cannot be mirrored.

Specify input if the port will monitor incoming traffic, output to monitor outgoing traffic, or both to monitor both types of traffic.

NOTE
In VPLS, when an unknown unicast traffic is handled, it uses the corresponding VLAN Forwarding ID to flood the packets to the VLAN domain which contains both the monitored port as well as the mirroring port. But in VLL, there is no such flood handling mechanism and hence, there is a discrepancy in the output of the **show statistic brief** command in terms of the **Packet Transmit** count on the mirroring port.

Displaying mirror and monitor port configuration

To display the inbound and outbound traffic mirrored to each mirror port, enter the following command at any level of the CLI.

```
NetIron# show monitor config
Monitored Port 3/1
    Input traffic mirrored to: 2/1
    Output traffic mirrored to: 1/1
Monitored Port 4/1
    Input traffic mirrored to: 1/2
    Output traffic mirrored to: 2/1
```

Syntax: show monitor config

To display the actual traffic mirrored to each mirror port, enter the following command at any level of the CLI.

```
NetIron# show monitor actual
Monitored Port 3/1
    Output traffic mirrored to: 1/1
Monitored Port 4/1
    Input traffic mirrored to: 1/2
```

Syntax: show monitor actual

This output displays the output traffic mirrored to mirror port 1/1 from port 3/1 and input traffic mirrored to mirror port 1/2 from port 4/1, which are explicitly configured.

ACL-based inbound mirroring

The Multi-Service IronWare software supports using an ACL to select traffic for mirroring from one port to another. Using this feature, you can monitor traffic in the mirrored port by attaching a protocol analyzer to it. The **acl-mirror-port** command is supported for POS interfaces. A user can configure a POS port to be monitored and then mirrored to an Ethernet port. Traffic cannot however be mirrored *to* a POS port.

Considerations when configuring ACL-based inbound mirroring

The following must be considered when configuring ACL-based inbound mirroring:

- Configuring a common destination ACL mirror port for all ports of a PPCR (see below)
- Support with ACL CAM sharing enabled (see below)
- The **mirror** and **copy-sflow** keywords are mutually exclusive on a per-ACL clause basis.
- ACL-based inbound mirroring and port-based inbound mirroring are mutually exclusive on a per-port basis.
- ACL-based mirroring must be configured at the LAG level for individual LAG member ports (see "Configuring ACL-based mirroring" on page 223).
- Configuring ACL-based mirroring at the port level on the primary port of a LAG mirrors all traffic on that LAG to the monitor port.

Configuring a Common Destination ACL mirror port for all ports of a PPCR

All ports using the same PPCR must have a common destination ACL mirror port when configuring ACL-based inbound mirroring. For Example, where ports 4/1 and 4/2 belong to the same PPCR, the following configuration that configures them with different destination ACL mirror ports will fail and generate an error message as shown.

```
NetIron(config)# interface ethernet 4/1
NetIron(config-if-e10000-4/1)# acl-mirror-port ethernet 6/1
NetIron(config-if-e10000-4/1)# interface ethernet 4/2
NetIron(config-if-e10000-4/2)# acl-mirror-port ethernet 6/2
Error: 4/2 and 4/1 should have the same ACL mirror port
```

Support with ACL CAM sharing enabled

For ACL CAM sharing to function, either one of the following conditions must be true:

- All ports that belong to a PPCR have the **acl-mirror-port** command configured to direct mirrored traffic to the same port.
- None of the ports that belong to the PPCR have the **acl-mirror-port** command configured.

ACL CAM sharing cannot function with the configuration shown in the following example because port 4/1 has ACL port mirroring configured and port 4/2 does not.

```
NetIron(config)# enable-acl-cam-sharing
NetIron(config)# interface ethernet 4/1
NetIron(config-if-e10000-4/1)# ip access-group 101 in
NetIron(config-if-e10000-4/1)# acl-mirror-port ethernet 6/1
NetIron(config-if-e10000-4/1)# interface ethernet 4/2
NetIron(config-if-e10000-4/2)# ip access-group 101 in
```

Configuring ACL-based inbound mirroring

The following sections describe how to configure ACL-based Inbound Mirroring on a PowerConnect router:

- Creating an ACL with a mirroring clause
- Applying the ACL to an interface
- Specifying a destination mirror port
- Specifying the desintation mirror port for physical ports
- Specifying the destination mirror port for a LAG
- Configuring ACL-based mirroring for ACLs bound to virtual interfaces
- Specifying the destination mirror port for IP receive ACLs

Creating an ACL with a mirroring clause

The **mirror** keyword in IPv4, L2 and IPv6 ACL clauses directs traffic that matches the clause criteria to be mirrored to another port. In the following examples, the ACL is used to direct IP traffic to a mirror port.

Example : ACL-based Mirroring Supported for IPv4 ACLs.

```
access-list 101 permit ip any any mirror
access-list 101 permit ip any any
```

Example : ACL-based Mirroring supported for IPv6 Inbound ACLs.

```
ipv6 access-list gem
   permit tcp 1000:1::/64 2000:1::/64 mirror
   permit udp 1000:1::/64 2000:1::/64 mirror
   permit icmp 1000:1::/64 2000:1::/64 mirror
   permit ipv6 any any
```

Example : ACL-based Mirroring supported for Layer-2 Inbound ACLs.

```
access-list 400 permit 0000.0000.0010 ffff.ffff.ffff 0000.0000.0020
           ffff.ffff.ffff.ffff any mirror
access-list 400 permit 0000.0000.0050 ffff.ffff.ffff 0000.0000.0020
           ffff.ffff.ffff.ffff any mirror
access-list 400 permit any any any
```

The **mirror** parameter directs selected traffic to the mirrored port. Traffic can only be selected using the **permit** clause. The mirror parameter is supported on rACLs.

NOTE
As with any ACL, the final clause must permit desired traffic to flow: be sure to add an appropriate **permit any any** clause to the end of any ACL intended to mirror (and not filter) traffic. Failure to include the **permit** clause will result in disruption of traffic through any interface to which the ACL is applied.

Applying the ACL to an interface

You must apply the ACL to an interface using the **ip access-group command** as shown in the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# ip access-group 101 in
```

Specifying the destination mirror port

You can specify physical ports or a LAG to mirror traffic from. The following sections describe how to perform each of these configurations.

Specifying the destination mirror port for physical ports

You must specify a destination port for traffic that has been selected by ACL-based Inbound Mirroring. This configuration is performed at the Interface Configuration of the port whose traffic you are mirroring. In the following example, ACL mirroring traffic from port 1/1 is mirrored to port 1/3.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# acl-mirror-port ethernet 1/3
```

You can also use the ACL-mirroring feature to mirror traffic from multiple ports to a single port using the Multiple Interface Configuration (MIF) mode as shown in the following example.

```
NetIron(config)# interface ethernet 1/1 to 1/2
NetIron(config-mif-e10000-1/1-1/2)# acl-mirror-port ethernet 1/3
```

Syntax: [no] acl-mirror-port ethernet [slot/port]

The [slot/port] variable specifies port that ACL-mirror traffic from the configured interface will be mirrored to.

Specifying the destination mirror port for a LAG

You can mirror the traffic that has been selected by ACL-based inbound mirroring from all ports in a LAG by configuring a destination (monitor) port for the LAG at the interface configuration level of the LAG's primary port. Configuring mirroring on the primary port of the LAG causes ACL-selected traffic from all ports in the LAG to be mirrored to the monitor port. For example, in the following configuration all traffic on LAG "mylag" will be mirrored to port 10/4:

```
NetIron(config)# lag mylag static
NetIron(config-lag-mylag)# ports ethernet 10/1 to 10/3
NetIron(config-lag-mylag)# primary-port 10/1
NetIron(config-lag-mylag)# deploy
NetIron(config-lag-mylag)# exit
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e1000-10/1)# acl-mirror-port ethernet 10/4
```

Syntax: [no] acl-mirror-port ethernet <*slot/port*>

The **ethernet** <*slot/port*> variable specifies the port that ACL-mirror traffic from the LAG will be mirrored to.

The following considerations apply when configuring ACL-based mirroring with LAGs:

- You must configure ACL-mirroring for an individual member port from the LAG configuration level (see "Configuring ACL-based mirroring" on page 223). Attempting to configure ACL-mirroring at the interface level for an individual member port will fail and display the following message.

  ```
  Error: please use  config level to configure ACL based mirroring on  port.
  ```

- If an individual port is configured for ACL-based mirroring, you cannot add it to a LAG. If you want to add it to a LAG, you must remove it from ACL-based mirroring first. Then you can add it to a LAG. It can then be configured for either ACL-based LAG mirroring or for mirroring an individual port within a LAG.

  If you attempt to add a port that is configured for ACL-based mirroring to a LAG, the following message will display.

  ```
  ACL port is configured on port 2/1, please remove it and try again.
   transaction failed:  Config Vetoed
  ```

- When a LAG with ACL-based mirroring configured on it is deleted or undeployed, the ACL-based mirroring configuration is removed from each of the individual ports that made up the LAG, including the primary port.

Configuring ACL-based mirroring for ACLs bound to virtual interfaces

For configurations that have an ACL bound to a virtual interface, you must configure the **acl-mirror-port** command on a port for each PPCR that is a member of the virtual interface. For example, in the following configuration ports 4/1 and 4/2 share the same PPCR while port 4/3 uses another PPCR.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# tagged ethernet 4/1 to 4/3
NetIron(config-vlan-10)# router-interface ve 10

NetIron(config)# interface ethernet 4/1
NetIron(config-if-e10000-4/1)# acl-mirror-port ethernet 5/1
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ip address 10.10.10.254/24
NetIron(config-vif-10)# ip access-group 102 in

NetIron(config)# access-list 101 permit ip any any mirror
```

In this configuration, the **acl-mirror-port** command is configured on port 4/1 which is a member of ve 10. Because of this, ACL-based mirroring will apply to VLAN 10 traffic that arrives on ports 4/1 and 4/2. It will not apply to VLAN 10 traffic that arrives on port 4/3 because that port uses a different PPCR than ports 4/1 and 4/2. To make the configuration apply ACL-based mirroring to VLAN 10 traffic arriving on port 4/3, you must add the following command to the configuration.

```
NetIron(config)# interface ethernet 4/3
NetIron(config-if-e10000-4/3)# acl-mirror-port ethernet 5/1
```

If the ve contains LAG ports, configuration of acl-mirror-port command on an individual LAG port will also apply to other LAG ports that are in the same PPCR. For example, in the following configuration the acl-mirror-port command is configured for LAG port 10/2, which is a member of ve.

```
NetIron(config)# lag mylag static
NetIron(config-lag-mylag)# ports ethernet 10/1 to 10/4
NetIron(config-lag-mylag)# primary-port 10/1
NetIron(config-lag-mylag)# deploy
NetIron(config-lag-mylag)# acl-mirror-port ether-port-monitored 10/2 ethe 11/3

NetIron(config)# vlan 10
NetIron(config-vlan-10)# tagged ethe 10/1 to 10/4
NetIron(config-vlan-10)# router-interface ve 10
```

The ACL-based mirroring will apply to VLAN 10 traffic incoming on ports 10/1 and 10/2 since they are in the same PPCR and are members of a virtual interface. However it will not apply to VLAN 10 traffic incoming on 10/3 and 10/4 since they are in a different PPCR. To apply ACL-based mirroring on VLAN 10 traffic incoming on 10/3 and 10/4, you will have to additionally configure the acl-mirror-port ethe-port-monitored 10/3 ethe 11/3 command under the LAG.

Specifying the destination mirror port for IP Receive ACLs

When specifying a destination port for IP Receive ACLs, you must configure the **acl-mirror-port** command on all ports supported by the same PPCR. For example, if you are using mirroring traffic for an rACL on a 4 x 10G interface module and you want to mirror traffic incoming on the first PPCR, you have to configure the **acl-mirror-port** command on both ports 1 and 2. If you want to mirror IP Receive ACL permit traffic incoming on all ports of the module, you have to configure the **acl-mirror-port** command on all ports of the module.

10G WAN PHY fault and performance management

This feature provides fault and performance management features like alarm detection, alarm generation and performance monitoring on 10 GbE WAN PHY interfaces. It only applies to 10 GbE interfaces configured in the WAN PHY mode.

Using this feature, you can gather fault and performance management information and display it for the current 15 minute interval or for any of the previous 15 minute intervals. In addition, this feature allows you to create a path trace between WAN PHY interfaces to ensure correct connection.

Setting a 10 GbE interface to WAN PHY mode

To set a 10 GbE interface to WAN PHY mode, use the following command.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e10000-3/1)# phy-mode wan
```

Syntax: [**no**] **phy-mode** { **wan** | **28k** }

The **wan** parameter sets the PHY mode to WAN.

The **28k** parameter sets the PHY mode to 28k (provided for compatibility with Bay Networks hardware).

The default setting is LAN PHY mode; to reset PHY mode to LAN, use the command **no phy-mode**.

Turning alarm interfaces on and off

When a 10 GbE port is to WAN PHY mode, alarm monitoring is set on by default. You can turn alarm monitoring off for an individual port as shown in the following.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e10000-3/1)# no alarm-monitoring
```

Syntax: [no] alarm-monitoring

Configuring path trace

You can configure a character string to be carried in the SONET overhead as a method of detecting mis-connection of ports between two routers connected over the WAN PHY. The routers compare the configured character string with the received character string to determine if the connection is valid.

You can configure an PowerConnect router with the character string "test1" using the following command.

```
NetIron(config)# overhead j0-transmit test1
```

Syntax: [no] overhead [j0-transmit <*string*> **] | [j1-transmit** <*string*>**]**

The <*string*> variable is the character string used to detect mis-connection of ports it must be configured with the same value on each side of the WAN PHY connection.

Displaying status of alarms on an interface

You can display the current status of WAN PHY alarms as shown in the following.

```
NetIron# show control1er curr15min e 3/1
----------------------------------------------------------
10g wan phy alarms statistics  - PORT e3/1
----------------------------------------------------------
ACTIVE ALAMRS : LOS
ACTIVE DEFECTS : LOF-S AIS-L AIS-P

Elapsed time [0 min 13 secs]
Format [alarm type = count]
FM-PARAMS
 Section
  LOS = 1  LOF = 1
Line
  AIS-L = 1 RDI-L = 0
Path
  AIS-P = 1  LOP = 0 PLM = 0 AIS-PFE = 0 PLM-PFE = 0
PM-PARAMS
 Section
  CV = 2  ES = 1  SES = 0  SEFS = 0
 Line
  CV = 3  ES = 1  SES = 0 UAS = 0
  CV-FE  = 0  ES-FE = 0  SES-FE = 0 UAS-FE = 0
 Path
   CV = 4  ES = 1  SES = 0 UAS = 0
   CV-FE  = 0  ES-FE = 0  SES-FE = 0 UAS-FE = 0
```

Syntax: show controller [curr15min <*port no***> | <***slot no***>] | [day <***port no***> | <***slot no***>] | [prev15min <** *interval***> <***port no***> | <***slot no***>]**

The **curr15min** parameter specifies that you want to display WAN PHY alarm and performance information for the current 15 minute interval for either the port number ***<portno>*** or slot number ***<slotno>*** specified.

The **day** parameter specifies that you want to display WAN PHY alarm and performance information for the current day for either the port number ***<portno>*** or slot number ***<slotno>*** specified.

The **prev15min** parameter specifies that you want to display WAN PHY alarm and performance information for the a past 15 minute interval as specified by the ***<interval>*** variable for either the port number ***<portno>*** or slot number ***<slotno>*** specified.
Possible values for ***<interval>*** are 1 - 31 and indicates which previous 15 minute interval you want to display information from. The closest previous interval is 1 and the farthest is 31.

The <*port no*> variable specifies the port that you want to display WAN PHY alarm and performance information for.

The <*slot no*> variable specifies the slot that you want to display WAN PHY alarm and performance information for.

This display shows the following information.

TABLE 28 WAN PHY display parameters

Parameter	Description.
Loss of signal (LOS)	LOS is raised when the synchronous signal (STS-N) level drops below the threshold at which a BER of 1 in 103 is predicted. It could be due to a cut cable, excessive attenuation of the signal, or equipment fault. LOS state clears when two consecutive framing patterns are received and no new LOS condition is detected.
Out of frame (OOF) alignment or SEF (Severely errored Frame)	OOF state occurs when four or five consecutive SONET frames are received with invalid (errored) framing patterns (A1 and A2 bytes). The maximum time to detect OOF is 625 microseconds. OOF state clears when two consecutive SONET frames are received with valid framing patterns.
Loss of frame (LOF) alignment	LOF state occurs when the OOF state exists for a specified time in milliseconds. LOF state clears when an in-frame condition exists continuously for a specified time in milliseconds.
Loss of pointer (LOP)	LOP state occurs when N consecutive invalid pointers are received or N consecutive new data flags (NDFs) are received (other than in a concatenation indicator), where N = 8, 9, or 10. LOP state clears when three equal valid pointers or three consecutive AIS indications are received. LOP can be identified as follows: • STS path loss of pointer (SP-LOP) · • VT path loss of pointer (VP-LOP)

TABLE 28 WAN PHY display parameters (Continued)

Parameter	Description.
Alarm indication signal (AIS)	The AIS is an all-ones characteristic or adapted information signal. It is generated to replace the normal traffic signal when it contains a defect condition in order to prevent consequential downstream failures being declared or alarms being raised. Line AIS defect is detected as a "111" pattern in bits 6, 7, and 8 of the K2 byte in five consecutive frames. Line AIS defect is terminated when bits 6, 7, and 8 of the K2 byte do not contain the code "111" for five consecutive frames. STS-Path AIS defect is detected as all ones in bytes H1 and H2 in three contiguous frames. STS-Path AIS defect is terminated when a valid STS Pointer is detected with the NDF set to "1001" (inverted) for one frame, or "0110" (normal) for three contiguous frames. AIS can also be identified as follows: • Line alarm indication signal (AIS-L) · • STS path alarm indication signal (SP-AIS) · • VT path alarm indication signal (VP-AIS)
Remote error indication (REI)	This is an indication returned to a transmitting node (source) that an errored block has been detected at the receiving node (sink). This indication was formerly known as far end block error (FEBE). REI can also be identified as the following: • Line remote error indication (REI-L) · • STS path remote error indication (REI-P) · • VT path remote error indication (REI-V)
Remote defect indication (RDI)	This is a signal returned to the transmitting terminating equipment upon detecting a loss of signal, loss of frame, or AIS defect. RDI was previously known as FERF. RDI can also be identified as the following: • Line remote defect indication (RDI-L) · • STS path remote defect indication (RDI-P) · • VT path remote defect indication (RDI-V)
B1 error (coding violation, CV)	Parity errors evaluated by byte B1 (BIP-8) of an STS-N are monitored. If any of the eight parity checks fail, the corresponding block is assumed to be in error.
B2 error (coding violation, CV)	Parity errors evaluated by byte B2 (BIP-24 x N) of an STS-N are monitored. If any of the N x 24 parity checks fail, the corresponding block is assumed to be in error.
B3 error (coding violation, CV)	Parity errors evaluated by byte B3 (BIP-8) of a VT-N (N = 3, 4) are monitored. If any of the eight parity checks fail, the corresponding block is assumed to be in error.
Errored Seconds (ES)	At each layer, an Errored Second (ES) is a second with one or more Coding Violations at that layer OR one or more incoming defects (e.g., SEF, LOS, AIS, LOP) at that layer has occurred. Far end - This is an indication returned to a transmitting node (source) that an errored block has been detected at the receiving node (sink). And Errored seconds - far end indicate this error in terms of errored seconds. ES can be identified as follows: • Section Errored seconds (ES-S)· • Line Errored seconds (ES-L), Line Errored seconds- Far end (ES-LFE)· • Path Errored seconds (ES-P), Path Errored seconds- Far end (ES-PFE)

TABLE 28 WAN PHY display parameters (Continued)

Parameter	Description.
Severely Errored seconds (SES)	At each layer, an Severely Errored Second (SES) is a second with x or more CVs at that layer, or a second during which at least one or more incoming defects at that layer has occurred. Values of x vary depending on the line rate and the Bit Error Rate. SES can be identified as follows: • Section Severely Errored seconds (SES-S). • Line Severely Errored seconds (SES-L), Line Errored seconds- Far end (SES-LFE). • Path Severely Errored seconds (SES-P), Path Errored seconds- Far end (SES-PFE)
Severely errored frame seconds (SEFS)	A Severely Errored Framing Second (SEFS) is a seconds with containing one or more SEF events. This counter is only counted at the Section Layer.
Unavailable seconds (UAS)	At the Line, Path, and VT layers, an unavailable second is calculated by counting the number of seconds that the interface is unavailable. At each layer, the SONET or SDH interface is said to be unavailable at the onset of 10 contiguous SESs. The 10 SESs are included in unavailable time. Once unavailable, the SONET or SDH interface becomes available at the onset of 10 contiguous seconds with no SESs. The 10 seconds with no SESs are excluded from unavailable time. With respect to the SONET or SDH error counts at each layer, all counters at that layer are incriminated while the SONET or SDH interface is deemed available at that layer. While the interface is deemed unavailable at that layer, the only count that is incriminated is UASs at that layer. UAS can be identified as follows: • Line Unavailable seconds (UAS-L), Line Unavailable seconds at far end (UAS-LFE). • Path Unavailable seconds (UAS-P), Path Unavailable seconds (UAS-PFE)

Wait for all cards feature

During a system reload, an Interface module comes up after it completes its initialization process. After an Interface module is up, its ports can come up. Since 10G modules have more packet processors to initialize, 1G ports are up earlier than 10G ports.

The **wait-for-all-cards** command directs all ports to come up at the same time. This is done by waiting for all Interface modules to come up first, before allowing for ports to come up. This command is shown in the following.

```
NetIron(config)# wait-for-all-cards
```

Syntax: [no] wait-for-all-cards

NOTE
With the **wait-for-all-cards** command enabled,10G ports will come up before 1G ports because Multi-Service IronWare software processes 10G port's state changes first.

Link fault signaling

You can enable link fault signaling on 1 or 10 gigabit interfaces. Link fault signalling (LFS) is a physical layer protocol that enables communication on a link between two 1 or 10 Gigabit Ethernet devices. When configured on a 1 or 10 Gigabit Ethernet port, the port can detect and report fault conditions on transmit and receive ports.

If LFS is configured on an interface, the following Syslog messages are generated when that interface go up or down or when the TX or RX fiber is removed from one or both sides of the link that has LFS configured:

- Interface ethernet1/1, state down - link down
- Interface ethernet1/1, state up

The Link and Activity LEDs on the module turn on when there is traffic transiting the link after the fiber is installed. To configure LFS, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/4
NetIron(config-if-e1000-1/4)# link-fault-signal
```

Syntax: [no] link-fault-signal

LFS is disabled by default.

To display if LFS is configured on an interface, enter the following command.

```
NetIron#show link-fault-signaling
Global Remote Fault : OFF
PORT #: REMOTE FAULT:
PORT 1/1: OFF
PORT 1/2: OFF
PORT 1/3: OFF
PORT 1/4: ON
PORT 1/5: OFF
PORT 1/6: OFF
PORT 1/7: OFF
PORT 1/8: OFF
PORT 1/9: OFF
```

Displaying and clearing remote fault counters

To display Remote Fault Notification (RFN) counters on 10GbE LAN physical interface, enter the following command .

```
NetIron# show remote-fault ethernet 1/1 to 1/4
Port      RFN Detected           Remote-fault count     time last RFN detected
----       -----------------    ----------------------  --------------------------
1/1            Yes                        15             Sep 29 22:03:03
1/2            No                          0                  -
1/3            No                         12             Aug 20 13:22:14
1/4            No                           0                -

** remote-fault counters are only supported for ports in LAN PHY mode on 10GE
modules. **
```

The example above displays remote fault notification counters with slot 1 as a 10GbE module, and ports 1/1, 1/2, 1/3, and 1/4 in LAN mode.

If the user enters a slot number that is not a 10 GbE port, or if any port in the port range is not a 10GbE port in LAN mode, the following error message is displayed.

```
NetIron# show remote-fault slot 3
remote-fault counters are only supported for ports in LAN PHY mode on 10 GE
modules.
```

To clear remote fault notification counters on a 10 GbE LAN physical interface, enter the following command.

```
NetIron#clear remote-fault slot 1
```

Syntax: show or clear remote-fault [ethernet <*slot#/port#*> **[to** <*slot#/port*>**] | slot** <*slot#*>**]**

You can display information for remote fault notification counters in a PowerConnect router by using the **show remote-fault** command without options,

Use the ethernet <*slot#/port#*> option to limit the display to a single ethernet port.

Use the to <*slot#/port*> option for a range of ports.

Use the slot <*slot#*> option to limit the display to a single slot.

The following table describes the output of the **show remote-fault** command

TABLE 29 Display of show remote-fault output

This field...	Displays...
Port	The <*port#*> variable specifies the port number for the interface module.
RFN Detected	The remote-fault notification is detected on a given interface. If “Yes” is displayed, then the remote-fault notification is detected on the given port at the time of inquiry. If “No” is displayed, then no remote-fault notification is detected on the given port at the time of inquiry.
Remote-fault count	The Remote-fault count displays the number of times the remote-fault notification is detected on a given interface. The number of times, include: • The time since the Interface Module was last powered on. • The time since the count was last cleared by the user. • The time since the interface was last configured as a LAN mode.
time last RFN detected	The time the remote-fault notification was last detected on a given interface.

Limits and restrictions

Current implementation with this feature has the following limitations:

- Works only on a 10GbE LAN interface. Information for ports in a WAN interface at the time of inquiry is not displayed. Information for a port that does not belong to 10 GbE module is not displayed.
- In a Management Module switchover state, the remote fault notification counts and detection time are maintained.
- The RFN counts of a port only reset to zero in the following conditions:
  - At slot initialization (power on).
  - When the **clear remote-fault** command is enabled on 10 GbE LAN interface.
  - When configuring a 10GbE port into WAN mode.

Local fault event detection and counters

Local fault event detection and counters are enabled for ports on 10GbE LAN physical interfaces when link fault signaling is enabled on the interface. If both local fault and remote fault events are detected on the same interface, then the remote fault event is reported.

If a port is down because of a local fault event, then a syslog message is generated to inform you of this event. The syslog message will display "(local fault)" in the dynamic log buffer. For more information on local fault syslog messages, refer to "Syslog messages system" on page 2264. The local fault event is also indicated in the **show interface** command. In the **show interface** command, the reason is displayed as "(local fault)". For more information on the **show interface** command, refer to "Displaying information for an interface for an Ethernet port" on page 121.

Displaying and clearing local fault counters

To display local fault counters on 10GbE LAN physical interface, enter the following command.

```
NetIron# show local-fault ethernet 2/1 to 2/2
Port  Local Fault Detected   Local-Fault Count     time last Local Fault detected
---- ---------------------  ------------------  -----------------------------
2/1        yes                 1                   Apr  3 18:06:28
2/2         yes                1                   Apr  3 18:06:28
```

The example above displays local fault counters for ports 2/1 and 2/2 in LAN physical mode.

To clear local fault counters on a 10 GbE LAN physical interface, enter the following command.

```
NetIron# clear local ethernet 2/1 to 2/2
Local-fault stats for port 2/1 is cleared.
Local-fault stats for port 2/2 is cleared.
```

The example above displays ports 2/1 and 2/2 cleared for local-fault statistics.

Syntax: show or clear local [ethernet <*slot#/port#*> **[to** <*slot#/port*>**] | slot** <*slot#*>**]**

You can display information for local fault counters in a PowerConnect router by using the **show local-fault** command without options,

Or use the ethernet <*slot#/port#*> option to limit the display to a single ethernet port.

Or use the to <*slot#/port*> option for a range of ports.

Use the slot <*slot#*> option to limit the display to a single slot.

The following table describes the output of the **show local-fault** command.

TABLE 30 Display of show local-fault output

This field...	Displays...
Port	The <*port#*> variable specifies the port number for the interface module.
Local Fault Detected	The local-fault is detected on a given interface. If "Yes" is displayed, then the local-fault event is detected on the given port at the time of inquiry. If "No" is displayed, then no local-fault event is detected on the given port at the time of inquiry.
Local-fault count	The local-fault count displays the number of times the local-fault event is detected on a given interface. The number of times, include: • The time since the Interface Module was last powered on. • The time since the count was last cleared by the user. • The time since the interface was last configured as a LAN mode.
time last Local Fault detected	The time the local-fault event was last detected on a given interface.

Displaying Network Processor statistics

The Network Processor (NP) counters track the packets and bytes that enter the ingress NP and exit the egress NP. Counts displayed are since the last time the **clear np statistics** command was issued.

The **show np statistics command** displays the NP statistics for all interface modules within a router or for an interface in a specified slot or port. A routed packet drop counter is added to the **show np statistics** command. For more information on the routed packet drop counter, see Table 31 on page 161. The following example displays an output from the **show np statistics** command on the Product Name.

Output of the Product Name is as follows.

```
NetIron MLXe# show np statistics ethernet 10/4
NP STATs IPC reply from slot 10 length =1608
Port 10/4 RX
NP Rx Raw Good Packet          = (115458)
NP Rx Forward Packet            = (115458)
NP Rx Discard Packet           = (0)
NP Rx Unicast Packet           = (44571)
NP Rx Broadcast Packet         = (0)
NP Rx Multicast Packet         = (70887)
NP Rx Send to TM Packet        = (115458)
NP Rx Bad Packet               = (0)
NP Rx Lookup Unavailable       = (0)
NP Rx ACL Drop                 = (0)
NP Rx Priority 0/1 Drop        = (0)
NP Rx Priority 2/3 Drop        = (0)
NP Rx Priority 4/5 Drop        = (0)
NP Rx Priority 6/7 Drop        = (0)
NP Rx Suppress RPF Drop        = (0)
NP Rx RPF Drop                 = (0)
NP Rx IPv4 Packet              = (0)
```

```
NP Rx IPv6 Packet              = (0)
NP Rx Route-only Drop          = (0)
NP Rx IPv6 Suppress RPF Drop   = (0)
NP Rx IPv6 RPF Drop Count      = (0)
NP Rx IPv4 Byte                = (0)
NP Rx IPv6 Byte                = (0)
NP Rx POS Ctrl Protocol Packet = (0)
NP Rx POS Link Drop            = (0)
NP Rx Routed Packet Drop     = (0)
Port 10/4 TX
NP Tx Sent to MAC Packet       = (1365518)
NP Tx Raw Good Packet          = (1365518)
NP Tx Source Port Supress Drop = (0)
NP Tx Bad Packet Count         = (0)
NP Tx Unicast Packet           = (1324427)
NP Tx Broadcast Packet         = (1)
NP Tx Multicast Packet         = (41090)
NP Tx IPX HW Forwarded Packet  = (41090)
NP Tx Receive from TM          = (1365518)
NP Tx ACL Drop                 = (0)
NP Tx IPv4 Packet              = (0)
NP Tx IPv6 Packet              = (0)
NP Tx IPv4 Byte                = (0)
NP Tx IPv6 Byte                = (0)
NP Tx POS Ctrl Protocol Packet = (0)
NP Tx POS Link Drop            = (0)
```

Syntax: show np statistics [ethernet <*slot/port*> **] [pos** <*slot/port*> **] [slot** <*slot-num*> **]**

You can use the **ethernet** or **pos** options and specify a <*slot/port*> variable to display NP statistics for an individual port.

You can use the slot option and specify a <*slot-num*> variable to display NP statistics for an individual interface module.

The **Tx** and **Rx** counters displayed are described in the following tables.

TABLE 31 Rx counters

Rx counter (per port)	Explanation
Rx Raw Good Packet	Number of good packets received from MAC
Rx Forward Packet	Number of forwarded packets by packet evaluation engine
Rx Discard Packet	Number of packets flagged for discard by packet evaluation engine
Rx Unicast Packet	Number of unicast (indicated by MAC DA) packets received
Rx Broadcast Packet	Number of broadcast (indicated by MAC DA) packets received
Rx Multicast Packets	Number of multicast (indicated by MAC DA) packets received
Rx Send to TM Packets	Number of packets sent to TM (= Rx Forward Packet – RL drops)
Rx Bad Packets	Number of packets that have MAC to NP interface errors
Rx Loopup Unavailable	Number of packets that have been dropped due to unavailability of the CAM interface for packet lookups
Rx ACL Drop	Drop counter for ACL drop on the ingress path
Rx Priority 0/1 Drop	Drop counter for ingress priority 0,1 packets
Rx Priority 2/3 Drop	Drop counter for ingress priority 2,3 packets

TABLE 31 Rx counters

Rx counter (per port)	Explanation
Rx Priority 4/5 Drop	Drop counter for ingress priority 4,5 packets
Rx Priority 6/7 Drop	Drop counter for ingress priority 6,7 packets
Rx Supress RPF Drop	Counter for suppressed RPF drops on the ingress path due to ACL override
Rx RPF Drop	Counter for RPF drop on the ingress
Rx IPv4 Packet	Raw packet count that have IPv4 EType (0x0800) and IP version of 0x4
Rx IPv6 Packet	Raw packet count that have IPv6 EType (0x86DD) and IP version of 0x6
Rx IPv6 Supress RPF Drop	Counter for IPv6 suppressed RFP drops on the ingress path due to ACL override
Rx IPv6 RPF Drop Count	Counter for IPv6 drop on the ingress
NP Rx Route-only Drop	Counts packets that have been dropped due to Route-Only configuration during MAC-DA processing.
Rx IPv4 Byte	Raw packet Bytes (+FCS) that have IPv4 etype (0x0800) and IP version equals 0x4
Rx IPv6 Byte	Raw packet Bytes (+FCS) that have IPv6 etype (0x86DD) and IP version equals 0x6
Rx POS Ctrl Protocol Packet	Number of control protocol packets received in POS mode
Rx POS Link Drop	Number of packets dropped due to link state in POS mode
Rx Routed Packet Drop	Number of received IPv4 or IPv6 routed packets that are dropped because the TTL is 0, or because routing is not enabled on the given virtual interface.

TABLE 32 Tx counters

TX counter (per port)	Explanation
Tx Sent to MAC Packet	Total number of packets sent to MAC for transmit
Tx Raw Good Packet	Total number of packets sent to egress processing logic that pass the initial length checks (min, max, offsets, bad packet etc.)
Tx Source Port Suppression Drop	Number of packets dropped because of transmit source port suppression
Tx Bad Packet Count	Total number of packets dropped in egress logic that fail the initial length checks (min, max, bad packet etc.)
Tx Unicast Packet	Number of unicast packets transmitted (from MAC DA)
Tx Broadcast Packet	Number of broadcast packets transmitted (from MAC DA)
Tx Multicast Packet	Number of multicast packets transmitted (from MAC DA)
Tx Receive From TM	Number of packets received from TM
Tx ACL Drop	Number of packets that have been dropped by the Outbound ACL Logic
Tx IPv4 Packet	Number of IPv4 packets transmitted out the port (Etype==0x0800 & IPver == 0x4)
Tx IPv6 Packet	Number of IPv6 packets transmitted out the port (Etype==0x86DD & IPver == 0x6)
Tx IPv4 Byte	Counts packet Bytes (+FCS) that have IPv4 etype (0x0800) and IP version equals 0x4
Tx IPv6 Byte	Counts packet Bytes (+FCS) that have IPv6 etype (0x86DD) and IP version equals 0x6
Tx POS Ctrl Protocol Packet	Number of control protocol packets sent in POS mode
Tx POS Link Drop	Number of packets dropped due to link state in POS mode

Relationships between some counters

Some of the values for counters displayed using the **show np statistics** command are the result of adding the contents of more than one counter. Table 33 and Table 34 describe these relationships between NP counters displayed.

TABLE 33 Relationships between RX counters

Total RX Packets	=	Rx Bad Packets + Rx Lookup Unavailable Packets + Rx Raw Good Packets
Rx Raw Good Packets	= = =	Rx Unicast Packets + Rx Multicast Packets + Rx Broadcast Packets Rx IPv4 Packets + Rx IPv6 Packets + Rx Other Packets Rx Forward Packets + Rx Discard Packets
Rx Forward Packets	=	Rx Sent to TM Packets + Rx RL drop packets
Rx Discard Packets	=	ACL drop + TTL drop + route-only drop + RPF drop + tag mismatch drop + VLAN blocking drop + segment filtering drop + drop by packet evaluation decisions +miscellaneous
Rx Priority Drops	=	RL drop + Rx Discard Packets

TABLE 34 Relationships between TX counters

Tx Raw Good Packets	= =	Tx Receive From TM Packets - Tx Bad Packets Tx Unicast Packets + Tx Broadcast Packets + Tx Multicast Packets + Tx Source Port Suppression Drop
Tx Sent to MAC	= =	Tx IPv4 Packets + Tx IPv6 Packets + Tx Others Tx Raw Good Packets – Tx Source Port Suppression Drop – Tx ACL drop - Tx RL Drop – Tx Multicast TTL drop

Clearing the NP statistics counters

You can clear the NP statistics counters for an entire router or selectively by port or slot using the **clear np statistics** command as shown in the following.

```
NetIron# clear np statistics
```

Syntax: clear np statistics [ethernet <*slot/port*> **] [pos** <*slot/port*> **] [slot** <*slot-num*> **]**

You can use the **ethernet** or **pos** options and specify a <*slot/port*> variable to clear NP statistics for an individual port.

You can use the **slot** option and specify a <*slot-num*> variable to clear NP statistics for an individual interface module.

Enabling the Foundry Discovery Protocol (FDP) and Reading Cisco Discovery Protocol (CDP) Packets

5

Overview

The following features are supported by NetIron MLX Series devices.

- Foundry Discovery Protocol (FDP)
- Reading Cisco Discovery Protocol (CDP)
- Displaying FDP Information
- Displaying CDP Information

NOTE
On platforms that support the Ethernet Service Instance (ESI) framework: FDP and CDP may be configured in the default ESI. FDP and CDP are not supported under user-defined ESIs.

This chapter discusses the following features:

- **Foundry Discovery Protocol (FDP)** – a protocol used by Dell devices to advertise themselves to other Dell devices
- **Cisco Discovery Protocol (CDP)** – a protocol used by Cisco devices to advertise themselves to other Cisco devices. Dell devices use this protocol to learn device and interface information for Cisco devices in the network

Using FDP

FDP enables Dell devices to advertise themselves to other Dell devices on the network. When you enable FDP on a Dell device, the device periodically advertises information including the following:

- Hostname (device ID)
- Product platform and capability
- Software version
- VLAN and Layer 3 protocol address information for the port sending the update.

A device running FDP sends FDP updates on Layer 2 to MAC address 01-E0-52-CC-CC-CC. Other devices listening on that address receive the updates and can display the information in the updates.

FDP is disabled by default.

NOTE
If FDP is not enabled on a device that receives an FDP update or the device is running a software release that does not support FDP, the update passes through the device at Layer 2. Also, FDP should not be used if VPLS is enabled.

Configuring FDP

The following sections describe how to enable FDP and how to change the FDP update and hold timers.

Enabling FDP globally

To enable a Dell device to globally send FDP packets, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# fdp run
```

Syntax: [no] **fdp run**

The feature is disabled by default.

Enabling FDP at the interface level

You can enable FDP at the interface level by entering the following commands.

```
NetIron(config)# int e 2/1
NetIron(config-if-e10000-2/1)# fdp enable
```

Syntax: [no] **fdp enable**

By default, the feature is enabled on an interface once FDP is enabled on the device.

Changing the FDP update timer

By default, a device enabled for FDP sends an FDP update every 60 seconds. You can change the update timer to a value from 5 – 900 seconds.

To change the FDP update timer, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# fdp timer 120
```

Syntax: [no] **fdp timer** <*secs*>

The <*secs*> parameter specifies the number of seconds between updates and can be from 5 – 900 seconds. The default is 60 seconds.

Changing the FDP hold time

By default, a device that receives an FDP update holds the information until one of the following events occurs:

- The device receives a new update.
- 180 seconds have passed since receipt of the last update. This is the hold time.

Once either of these events occurs, the device discards the update.

To change the FDP hold time, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# fdp holdtime 360
```

Syntax: [**no**] **fdp holdtime** <*secs*>

The <*secs*> parameter specifies the number of seconds a device that receives an FDP update can hold the update before discarding it. You can specify from 10 – 255 seconds. The default is 180 seconds.

Displaying FDP information

You can display the following FDP information:

- FDP entries for neighbors
- Individual FDP entries
- FDP information for an interface on the device you are managing
- FDP packet statistics

NOTE
If the device has intercepted CDP updates, the CDP information is also displayed.

Displaying neighbor information

To display a summary of all the neighbors that have sent FDP updates to this device, enter the following command.

```
NetIronA# show fdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
(*) indicates a CDP device

   Device ID      Local Int    Holdtm Capability Platform    Port ID
   -------------- ------------ ------ ---------- ----------- -------------
   NetIronB    Eth 2/9      178    Router     NetIron Rou Eth 2/9
```

Syntax: **show fdp neighbor** [**ethernet** <*slot*>/<*portnum*>] [**detail**]

The **ethernet** <*slot*>/<*portnum*> parameter lists the information only for updates received on the specified interface.

The **detail** parameter lists detailed information for each device.

The **show fdp neighbor** command, without optional parameters, displays the following information.

TABLE 35 Summary FDP and CDP neighbor information

This line...	Displays...
Device ID	The hostname of the neighbor.
Local Int	The interface on which this device received an FDP or CDP update for the neighbor.
Holdtm	The maximum number of seconds this device can keep the information received in the update before discarding it.
Capability	The role the neighbor is capable of playing in the network.
Platform	The product platform of the neighbor.
Port ID	The interface through which the neighbor sent the update.

To display detailed information, enter the following command.

```
NetIronA# show fdp neighbor detail
Device ID: PowerConnectB configured as default VLAN1, tag-type8100
Entry address(es):
Platform: PowerConnect Router,  Capabilities: Router
Interface: Eth 2/9
Port ID (outgoing port): Eth 2/9 is TAGGED in following VLAN(s):
 9 10 11
Holdtime : 176 seconds
Version :
Brocade Router, IronWare Version 07.6.01b1T53 Compiled on Aug 29
2002 at 10:35:21 labeled as B2R07601b1
```

The **show fdp neighbor detail** command displays the following information.

TABLE 36 Detailed FDP and CDP neighbor information

This line...	Displays...
Device ID	The hostname of the neighbor. In addition, this line lists the VLAN memberships and other VLAN information for the neighbor port that sent the update to this device.
Entry address(es)	The Layer 3 protocol addresses configured on the neighbor port that sent the update to this device. If the neighbor is a Layer 2 Switch, this field lists the management IP address.
Platform	The product platform of the neighbor.
Capabilities	The role the neighbor is capable of playing in the network.
Interface	The interface on which this device received an FDP or CDP update for the neighbor.
Port ID	The interface through which the neighbor sent the update.
Holdtime	The maximum number of seconds this device can keep the information received in the update before discarding it.
Version	The software version running on the neighbor.

Displaying FDP entries

To display the detailed neighbor information for a specific device, enter a command such as the following.

```
NetIronA# show fdp entry NetIronB
Device ID: NetIronB configured as default VLAN1, tag-type8100
Entry address(es):
Platform: NetIron Router,  Capabilities: Router
Interface: Eth 2/9
Port ID (outgoing port): Eth 2/9 is TAGGED in following VLAN(s):
 9 10 11
Holdtime : 176 seconds
Version :
Brocade Router, IronWare Version 07.6.01b1T53 Compiled on Aug 29
2002 at 10:35:21 labeled as B2R07601b1
```

Syntax: **show fdp entry** * | <*device-id*>

The * | <*device-id*> parameter specifies the device ID. If you enter *, the detailed updates for all neighbor devices are displayed. If you enter a specific device ID, the update for that device is displayed. For information about the display, refer to Table 36 on page 168.

Displaying FDP information for an interface

To display FDP information for an interface, enter a command such as the following.

```
NetIronA# show fdp interface ethernet 2/3
FastEthernet2/3 is up, line protocol is up
  Encapsulation ethernet
  Sending FDP packets every 5 seconds
  Holdtime is 180 seconds
```

This example shows information for Ethernet port 2/3. The port sends FDP updates every 5 seconds. Neighbors that receive the updates can hold them for up to 180 seconds before discarding them.

Syntax: show fdp interface [**ethernet** <*slot*>/<*portnum*>]

The **ethernet** <*slot*>/<*portnum*> parameter lists the information only for the specified interface.

Displaying FDP and CDP statistics

To display FDP and CDP packet statistics, enter the following command.

```
NetIronA# show fdp traffic
CDP/FDP counters:
  Total packets output: 6, Input: 5
  Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
  No memory: 0, Invalid packet: 0, Fragmented: 0
  Internal errors: 0
```

Syntax: show fdp traffic

Clearing FDP and CDP information

You can clear the following FDP and CDP information:

- Information received in FDP and CDP updates
- FDP and CDP statistics

The same commands clear information for both FDP and CDP.

Clearing FDP and CDP neighbor information

To clear the information received in FDP and CDP updates from neighboring devices, enter the following command.

```
NetIron# clear fdp table
```

Syntax: clear fdp table

NOTE
This command clears all the updates for FDP and CDP.

Clearing FDP and CDP statistics

To clear FDP and CDP statistics, enter the following command.

```
NetIron# clear fdp counters
```

Syntax: clear fdp counters

Reading CDP packets

Cisco Discovery Protocol (CDP) packets are used by Cisco devices to advertise themselves to other Cisco devices. By default, a device forwards these packets without examining their contents. You can configure a device to intercept and display the contents of CDP packets. This feature is useful for learning device and interface information for Cisco devices in the network.

Dell devices support intercepting and interpreting CDP version 1 and 2 packets.

NOTE
The device can interpret only the information fields that are common to both CDP version 1 and CDP version 2.

NOTE
When you enable interception of CDP packets, the device drops the packets. As a result, Cisco devices will no longer receive the packets.

Enabling interception of CDP packets globally

To enable the device to intercept and display CDP packets, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# cdp run
```

Syntax: [no] cdp run

The feature is disabled by default.

Enabling interception of CDP packets on an interface

You can disable and enable CDP at the interface level.

You can enter the following commands.

```
NetIron(config)# int e 2/1
NetIron(config-if-e10000-2/1)# cdp enable
```

Syntax: [no] cdp enable

By default, the feature is enabled on an interface once CDP is enabled on the device.

Displaying CDP information

You can display the following CDP information:

- Cisco neighbors
- CDP entries for all Cisco neighbors or a specific neighbor
- CDP packet statistics

Displaying neighbors

To display the Cisco neighbors the device has learned from CDP packets, enter the following command.

```
NetIron# show fdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
(*) indicates a Cisco device

   Device ID      Local Int    Holdtm Capability Platform    Port ID
   -------------- ------------ ------ ---------- ----------- -------------
(*)Router         Eth 1/1      124    R          cisco RSP4
FastEthernet5/0/0
```

Syntax: show fdp neighbors [**detail** | **ethernet** <*portnum*>]

To display detailed information for the neighbors, enter the following command.

```
NetIron# show fdp neighbors detail
Device ID: Router
Entry address(es):
  IP address: 207.95.6.143
Platform: cisco RSP4,  Capabilities: Router
Interface: Eth 1/1,  Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 150 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1,  RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

To display information about a neighbor attached to a specific port, enter a command such as the following.

```
NetIron# show fdp neighbors ethernet 1/1
Device ID: Router
Entry address(es):
  IP address: 207.95.6.143
Platform: cisco RSP4,  Capabilities: Router
Interface: Eth 1/1,  Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 127 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1,  RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

Displaying CDP entries

To display CDP entries for all neighbors, enter the following command.

```
NetIron# show fdp entry *
Device ID: Router
Entry address(es):
  IP address: 207.95.6.143
Platform: cisco RSP4,  Capabilities: Router
Interface: Eth 1/1,  Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 124 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1,  RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

Syntax: show fdp entry * | <*device-id*>

For example, to display CDP entries for a specific device, specify the device ID.

```
NetIron# show fdp entry Router1
Device ID: Router1
Entry address(es):
  IP address: 207.95.6.143
Platform: cisco RSP4,  Capabilities: Router
Interface: Eth 1/1,  Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 156 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1,  RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

Displaying CDP statistics

To display CDP packet statistics, enter the following command.

```
NetIron# show fdp traffic
CDP counters:
  Total packets output: 0, Input: 3
  Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
  No memory: 0, Invalid packet: 0, Fragmented: 0
```

Syntax: show fdp traffic

Clearing CDP information

You can clear the following CDP information:

- Cisco Neighbor information
- CDP statistics

To clear the Cisco neighbor information, enter the following command.

```
NetIron# clear fdp table
```

Syntax: clear fdp table

To clear CDP statistics, enter the following command.

```
NetIron# clear fdp counters
```

Syntax: clear fdp counters

Using a Redundant Management Module 6

Overview

The following Redundant Management Module features are supported by NetIron MLX Series devices.

- Management Module Switchover
- Default Active Chassis Slot
- Synchronization Between Active and Standby Management Modules
- Manually Switching Over to the Standby Management Module
- Monitoring Management Module Redundancy
- Flash Memory and PCMCIA Flash Card File Management
- Option to delete old file first upon image download if MP Flash full
- Enhanced Syslog for Module State Changes

You can install a redundant management module in slot M1 or M2 of a PowerConnect chassis. (By default, the system considers the module in slot M1 to be the active management module and the module in slot M2 to be the redundant, or standby module. If the active module becomes unavailable, the standby module automatically takes over management of the system.

This chapter describes the redundant management module, how it works with the active module, and how to configure and manage it.

How management module redundancy works

This section explains the following:

- How management module redundancy works under normal operating conditions
- Events that cause a standby management module to assume the role of the active module (switchover)
- System implications when a switchover occurs

Management module redundancy overview

When you apply power to or reload a PowerConnect router with two management modules installed, by default, the management module in slot M1 becomes the active module and the module in slot M2 becomes the standby module. (You can change the default active slot from M1 to M2 using the **active-management** command. Refer to "Changing the default active chassis slot" on page 178.)

After the active and standby modules are determined, both modules boot from the source specified for the active module. The active module can boot from the following sources:

- The flash memory on the active management module
- A PCMCIA flash card in a PCMCIA slot on the active management module.

Once the modules boot, the system compares the flash code and system-config files on the standby module to the files on the active module. If the files are not the same, the files on the standby module are synchronized with those on the active module.

During normal operation, the active module handles tasks such as obtaining network topology and reachability information and determining the best paths to known destinations. The active module also monitors the standby module.

The standby module functions in an active standby mode. Configuration changes made from the CLI to the active management module are also written to the standby management module even if they are not written to flash memory. Synchronizing the system-config and running-config files on both modules allows the standby module to assume the role of active module seamlessly, if necessary.

The interface modules are not reset, and continue to forward traffic while the standby management module takes over operation of the system. The new now-active management module receives updates from the interface modules and sends verification information to the interface modules to ensure that they are synchronized. If the new active management module becomes out of sync with an interface module, information on the interface module may be overwritten, which can cause an interruption of traffic forwarding. An out of sync state should only occur if there is a layer 3 topology change elsewhere in the network during the management failover. PowerConnect devices support Layer 3 hitless failover with restart for high-availability routing in protocols such as BGP and OSPF. With these high-availability features enabled, when a router experiences a failover or restart, forwarding disruptions are minimized, and route flapping diminished to provide continuous service.

Management module switchover

The following events cause the standby management module to become the active module, which is called a ***switchover***:

- The active module becomes unavailable
- You perform a manual switchover
- You remove and replace the active management module

The following sections explain how the switchover occurs for each event.

Unavailable active module

The following events cause an active module to become unavailable and a switchover to occur:

- An active module experiences a problem significant enough to cause a reset of the module
- The active module loses power

Before a switchover occurs, the active module resets itself and sends an interrupt signal to the standby module. The standby module then becomes the active module and the interface modules continue to forward traffic.

The new active module begins to manage the system. When the original active module becomes available again or is replaced, it assumes the role of standby module.

Manual switchover

In some situations, you may want to manually switch the active module to the standby module. You can perform a manual switchover using the **switchover** command. For information about performing this task, refer to "Manually switching over to the standby management module" on page 181.

When the switchover occurs, the standby module becomes active and the active module becomes standby.

Removal and replacement of a management module

For information about how to remove and replace a management module, refer to "Replacing a Management Module" in the *PowerConnect B-MLXe Hardware Installation Guide.*

This section explains how management module redundancy is affected when you remove and replace an active or standby management module.

Removal and replacement of an active management module

If you remove the active management module, the standby module automatically assumes the active role. When you insert a replacement module in the slot from which the original active module was removed, the replacement module assumes the standby role. This module boots from a source specified for the active module, for example:

- The flash memory on the active management module
- A PCMCIA flash card installed in the active management module

When the replacement module boots, the system compares the flash code and system-config files on the standby module to the files on the active module. If differences exist, the files on the standby module are synchronized to match those on the active module.

Removal and replacement of a standby management module

You can remove a standby management module without causing a switchover to occur. The active module continues to function normally. When the new module is installed, it assumes the role of standby, and boots from a source specified for the active module, for example:

- The flash memory on the active management module
- A PCMCIA flash card installed in the active management module

The system compares the flash code and system-config files on the replacement module to the files on the active module. If differences exist, the files on the standby module are synchronized to match those on the active module.

Switchover implications

When a switchover occurs between the active and standby modules, the following areas may be affected:

- Management sessions
- Syslog and SNMP traps
- BGP Peer Notification – Described in "BGP4 Peer notification during a management module switchover" on page 1005

The following sections explain the implications for these areas.

Management sessions

You can establish management sessions using the management port on the active management module. If a switchover occurs, the management port on the original active module shuts down and all open CLI, Web Management Interface, and SNMP Network Manager sessions with that port close. You can open new sessions with the new active module, if this module has the same management port connections.

For example, if you were accessing the Web Management Interface through a PC connected to the original active management port, you can open a new session if a PC is connected to the new active management port. Open a new session using the same IP address you used before the switchover. (If a switchover occurs, the IP address you configured on the original active module is automatically assumed by the new active module.)

Syslog and SNMP traps

When a switchover occurs, the PowerConnect system sends a Syslog message to the local Syslog buffer and to the Syslog server, if you have configured the system to use one. The system also sends an SNMP trap to the receiver, if one is configured.

When system power is restored, or the system is reset normally, a cold start message and trap are sent. However, if the system is reset as the result of switchover to the standby management module, the system sends a warm start message and trap.

Management module redundancy configuration

Configuring management module redundancy consists of performing one optional task (changing the default active chassis slot) as described in the following section.

Changing the default active chassis slot

By default, the PowerConnect system considers the module installed in slot M1 to be the active management module. However, you can change the default active chassis slot to M2 using the **active-management** command.

The **active-management** command determines which management module will become active after a power cycle. By default, the top management module of the PowerConnect B-MLXe-16, PowerConnect B-MLXe-4 and PowerConnect B-MLXe-8 become active after a power cycle. This information is stored in the chassis's backplane EPROM and not in the configuration file.

To change the default active chassis slot from the default state of M1 to M2, enter the following commands.

```
NetIron(config)# redundancy
NetIron(config-redundancy)# active-management mgmt-2
```

Syntax: active-management <*mgt-module*>

The <*mgt-module*> parameter specifies the management module, either mgmt-1 or mgmt-2.

NOTE
This configuration has no effect on the **reload** and **boot** commands. it only applies to the power cycle when both management modules are installed in a chassis.

Managing management module redundancy

You can perform the following management tasks related to management module redundancy for PowerConnect devices:

- Perform immediate synchronization of files
- Perform a manual switchover to the standby module
- Reboot the standby module

File synchronization between active and standby management modules

Each active and standby management module contains the following files that can be synchronized between the two modules:

- **Flash code** – The flash code can include the following files:
  - monitor, which contains the Real Time Operating System (RTOS) for the management module
  - primary, which contains the primary Multi-Service IronWare image for the management module
  - secondary, which contains the secondary Multi-Service IronWare image for the management module

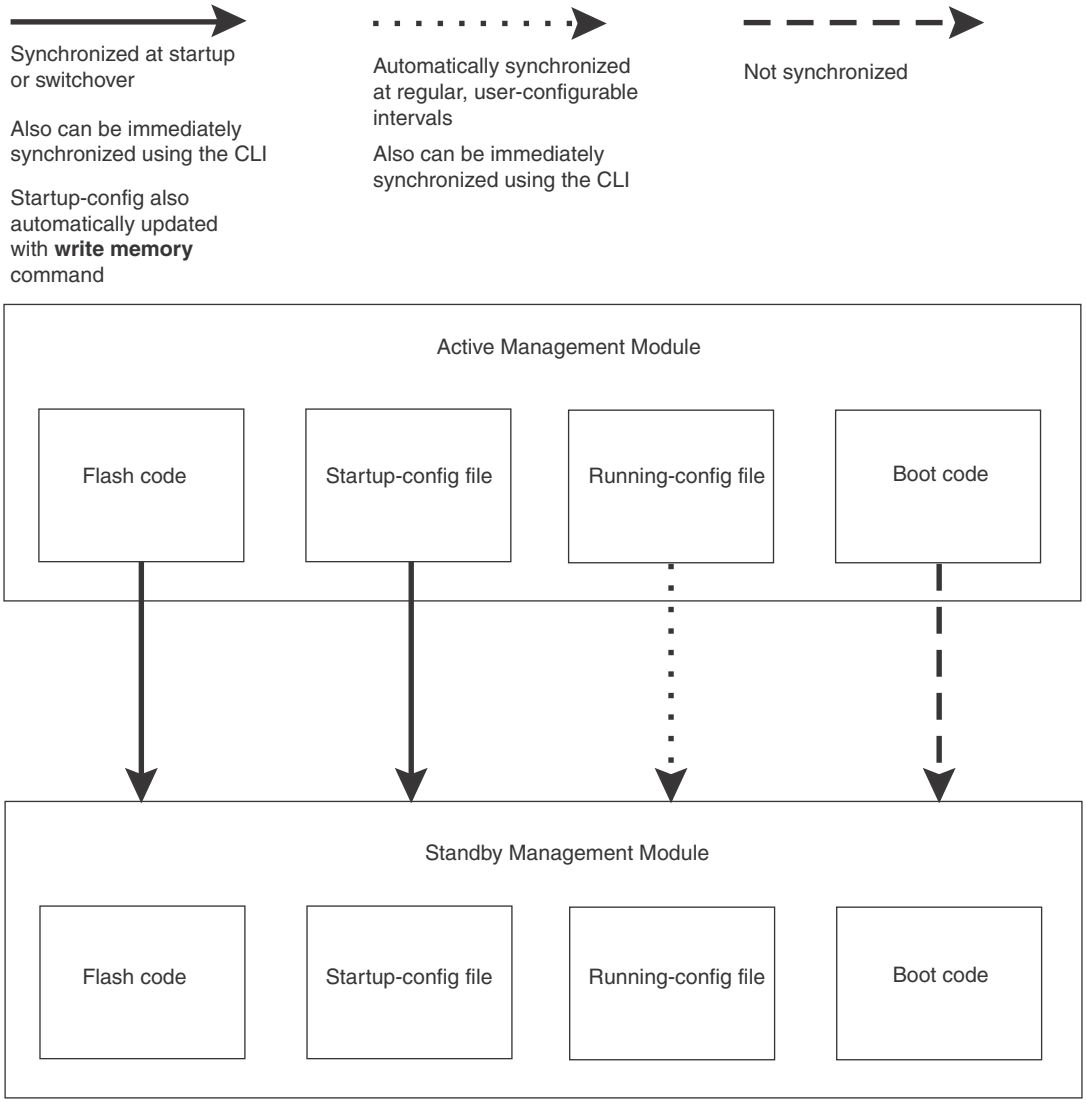
  A PowerConnect Multi-Service IronWare image contains layer 1 – 3 software used by the management module.

  During startup or switchover, the flash code on the active module is compared to the flash code on the standby module. If the files differ, the files on the standby module are synchronized to the files on the active module. If you update the flash code on the active module, the flash code on the standby module is automatically synchronized (without comparison) to the new file on the active module.
- **System-config file** – The flash code includes the system-config file. During startup or switchover, the system-config file on the active module is compared to the system-config file on the standby module. If the files are different, the system-config file on the standby module is synchronized with that of the active module. When you save changes to the system-config file on the active module, the system-config file on the standby module is automatically (without comparison) synchronized to match the system-config file on the active module.
- **Running-config** – The running-config file resides in the PowerConnect system memory, and is automatically synchronized (without comparison) between the active and the standby module at regular intervals. The default interval is 7 seconds.

Each active and standby management module also includes boot code that is run when a module boots. The boot code resides in the boot flash of each module. Boot code is not synchronized between the active and standby modules, which allows the system to use an older version of boot code on the standby module if desired.

Figure 2 shows how the files are synchronized between the active module and the standby module.

FIGURE 2 Active and standby management module file synchronization



The PowerConnect system allows you to perform the following file synchronization tasks:

- Compare files on the active module with files on the standby module and immediately synchronize any files that are different.
- Immediately synchronize all files between the active and standby modules.

The following sections explain how to perform these tasks.

Comparing and synchronizing files

You can initiate a comparison of the flash code, system-config, and running-config files on the active management module with these files on the standby module and synchronize the files immediately if differences exist. When you synchronize the files, the active module files are copied to the standby module, replacing the standby module files.

To compare and immediately synchronize files between the active and standby modules, enter the following command at the Privileged EXEC level.

```
NetIron# sync-standby
```

Synchronizing files without comparison

You can synchronize the flash code, system-config file, and running-config file immediately without comparison. When you synchronize the files, active module files are copied to the standby module, replacing the files on the standby module.

To immediately synchronize the files between the active and standby modules, enter the following command at the Privileged EXEC level.

```
NetIron# force-sync-standby
```

Manually switching over to the standby management module

You can cause the PowerConnect system to switch over to the standby module (and thus make it the active module). Enter the **switchover** command at the Privileged EXEC level.

```
NetIron# switchover
```

You are presented with the question **"Are you sure?"** after the switchover command is executed. At this question, you can either type **y** to proceed with the switchover or type **n** to abort the switchover.

The following is an example of the new switchover procedure.

```
NetIron#switchover
Are you sure? (enter 'y' or 'n'): y
```

NOTE
The switchover command should not be used immediately after downloading new code to the PowerConnect systems with redundant management modules.

Upon switchover to the standby management module, a software image is downloaded. The following error message is displayed on the console:

```
Warning: There is an outstanding software download. Do you want to continue?
(enter 'y' or 'n')
```

If you want to continue with downloading the image, enter yes. If you do not want to download the image, enter no.

Rebooting the active and standby management modules

You can reboot management modules, while maintaining the active and standby roles, using the **boot system** or **reload** commands. You can also reboot the standby module only, maintaining the standby role, using the **reboot-standby** command.

For example, to reboot the active and standby management modules from the primary PowerConnect Multi-Service IronWare image in the management module flash memory, enter the following command at the Privileged EXEC level.

```
NetIron# boot system flash primary
NetIron# Are you sure? (enter 'y' or 'n'): y
```

Syntax: **[no] boot system bootp | [flash primary | flash secondary] | slot** <*number*> <*filename*> | **tftp** <*ip-address*> <*filename*>

The **flash primary** keyword specifies the primary PowerConnect Multi-Service IronWare image in the management module flash memory. The **flash secondary** keyword specifies the secondary PowerConnect Multi-Service IronWare image in the flash memory.

For the <*number*> parameter, specify 1 for PCMCIA slot 1 on the active management module and 2 for PCMCIA slot 2 on the active management module. For the <*filename*> parameter, specify the name of the image on the PCMCIA flash card.

The **tftp** keyword directs the PowerConnect router to boot from an PowerConnect Multi-Service IronWare image on a TFTP server located at <*ip-address*> with the specified <*filename*>.

For example, to reboot the active and standby management modules, enter the following command at the Privileged EXEC level.

```
NetIron# reload
```

To reboot the standby module only, enter the following command at the Privileged EXEC level.

```
NetIron# reboot-standby
```

Upon rebooting the active and standby management module, a software image is downloaded. The following error message is displayed on the console:

```
Warning: There is an outstanding software download. Do you want to continue?
(enter 'y' or 'n')
```

If you want to continue with downloading the image, enter yes. If you do not want to download the image, enter no.

Monitoring management module redundancy

You can monitor the following aspects of management module redundancy:

- The status of the management modules (if a module is in active or standby mode)
- The switchover history for the management modules

The following sections explain how to monitor the management modules.

Determining management module status

You can determine the status of a management module in the following ways:

- **LEDs** – LEDs on the management module indicate whether a module is active or standby, and if the module has power.
- **Module information in software** – The module information displayed by the software indicates whether a module is active or standby.

Status LED

You can determine which management module is currently active and which is standby by observing the Active LED on each module. If this LED is on (green), the module is the active module. If this LED is off, the module is the standby module.

You can also observe the Pwr LED on each module. If this LED is on (green), the module is receiving power. If the LED is off, the module is not receiving power. (A module without power will not function as either the active or standby module.)

For information about what to do if these LED indicators are not what you expect, refer to the *PowerConnect B-MLXe Hardware Installation Guide.*

Software

To display the status of the management modules using the software, enter the following command at any level.

```
NetIron# show module
        Module                              Status      Ports  Starting MAC
M1 (left): NI-MLXe-MR Management Module     Active
M2 (right): NI-MLXe-MR Management Module    Standby (Ready)
)
...
```

The Status column indicates the module status. The management module status can be one of the following:

- **ACTIVE** – Current active management module
- **STANDBY** – Current standby management module.

  The status of the standby module can be one of the following:

  - **Init** – Currently initializing as the standby module
  - **Ready** – Ready to take over as the active module, if necessary
  - **Wait** – Waiting for boot information from the active management module
  - **Sync** – Active module is currently synchronizing files on the standby module

Monitoring the status change of a module

The NetIron system now logs the status change of a module. The status change of a module is logged when the module becomes:

- **Up or Ready** - The module is running or ready to run.
- **Down** - The module is not running normally.

Upon the status change of a module, a message is logged in the syslog memory. At the CLI level, type the **show log** command to view the logged messages.

The following example displays a syslog message on an Interface Module in the Down state.

```
Feb  5 12:16:17:N:System: Module down in slot 1, reason REBOOTED. Error Code 0
```

The following example displays a syslog message on a Standby Management Module in the Down state.

```
Feb  5 14:38:58:N:System: Standby Management Module was down, reason Heartbeat
Loss. Error Code 5
```

Displaying temperature information

All management, interface and switch fabric modules contain temperature sensors. By default, the PowerConnect system polls module temperature every 60 seconds. You can display the current temperature of the modules by entering either of the following commands:

- show chassis
- show temperature

For information about these commands, refer to the *PowerConnect B-MLXe Hardware Installation Guide.*

Displaying switchover information

You can display the following information about a switchover:

- Redundancy parameter settings and statistics, including the number of switchovers that have occurred
- System log or traps logged on an SNMP trap receiver, including Information about whether a switchover has occurred.

To view the redundancy parameter settings and statistics, enter the following command at any level of the CLI.

```
NetIron# show redundancy
=== MP Redundancy Settings ===
Default Active Slot = M1 (upper)
Running-Config Sync Period = 7 seconds

=== MP Redundancy Statistics ===
Current Active Session:
Active Slot=M2(lower),Standby Slot=M1(upper)(Ready State), Switchover Cause = No
Switchover
Start Time = 1900-0-0  0:6:21 (Monday)

Previous Active Session #1:
Active Slot=M1(upper), Standby Slot=M2(lower), Switchover Cause = MP Upgrade to
Ver3.7.0T163
Start Time = 1900-0-0  0:3:4 (Monday), End Time = 1900-0-0  0:6:21 (Monday)

Previous Active Session #2:
Active Slot = M2 (lower), Standby Slot = M1(upper), Switchover Cause = Active
Rebooted
Start Time = 1900-0-0  0:1:1 (Monday), End Time = 1900-0-0  0:3:4 (Monday)

Previous Active Session #3:
Active Slot = M1 (upper), Standby Slot = M2(lower), Switchover Cause = MP Upgrade
to Ver3.7.0T163
Start Time = 2036-2-6  6:43:54 (Wednesday), End Time = 1900-0-0  0:1:1 (Monday)
...
```

This output displays that the default active chassis slot is configured as slot M1 and the automatic synchronization interval is configured for 7 seconds. It also displays that in the current active session, the module installed in M2 is the active module, the module installed in M1 is the standby module, which is in Ready state, and no switchovers have occurred.

However, in three previous sessions, switchovers occurred. In sessions #1 and #3, the switchovers occurred because the software was upgraded to "Ver3.7.0T163". In session #2 the switchover occurred because the active module was rebooted. In sessions #1 and #3, the modules installed in M1 were the active modules, while the modules installed in M2 were the standby modules. In session #2, the module installed in M2 was the active module, while the module installed in M1 was the standby module.

To view the system log or traps logged on an SNMP trap receiver, enter the following command at any level.

```
NetIron# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 24 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Sep 28 11:31:25:A:Power Supply 1, 1st left, not installed
Sep 28 11:31:25:A:Power Supply 3, middle left, not installed
Sep 28 11:31:25:A:Power Supply 4, middle right, failed
Sep 28 11:31:25:A:Power Supply 5, 2nd right, not installed

Dynamic Log Buffer (50 lines):
Sep 27 18:06:58:I:Interface ethernet6/2, state up
Sep 27 18:06:57:I:Interface ethernet3/2, state up
Sep 27 15:39:42:I:Interface ethernet3/2, state up
Sep 27 15:39:42:I:Interface ethernet6/2, state up
...
Sep 27 14:23:45:N:Module up in slot 6
Sep 27 14:23:45:N:Module up in slot 3
Sep 27 14:23:27:A:Management module at slot 9 state changed from standby to
active
```

This output indicates that one switchover occurred.

Flash memory and PCMCIA flash card file management commands

The PowerConnect system supports file systems in the following locations:

- Flash memory on the management module
- A PCMCIA flash card inserted in management module slots 1 or 2

Table 37 outlines the root directory for each file system.

TABLE 37 PowerConnect file system root directories

File system	Root directory
Flash memory	/flash/
PCMCIA flash card in slot 1	/slot1/
PCMCIA flash card in slot 2	/slot2/

This section describes commands that manage the files in flash memory and on the flash cards. Use the file management commands to perform the following tasks:

- Format a flash card
- Determine the current management focus
- Switch the management focus
- Display a directory of files
- Display the contents of a file
- Display the hexadecimal output of a file
- Create a subdirectory
- Remove a subdirectory
- Rename a file
- Change the read-write attribute of a file
- Delete a file
- Recover (undelete) a file
- Append one file to another (join two files)
- Perform copy operations using the **copy** command
- Perform copy operations using the **cp** command
- Load the system software from flash memory, a flash card, or other sources during system reboot
- Change the save location of the startup-config file from the default location (flash memory) to a flash card in slot 1 or 2

You can access all file management commands at the Privileged EXEC level of the CLI.

CAUTION

Do not add or remove a flash card while a file operation involving the slot where the flash card is installed is in progress. Doing so can result in corruption of the flash card. If this occurs, you may need to reformat the flash card to make it usable again. Reformatting erases all data stored on the card.

Verifying available flash space on the management module before an image is copied

The Management Module of the NetIron system accommodates 32 MB of flash space. However, as the size of the Interface Module, Management Module, and FPGA images increase, the Management Module flash may not have enough space to accommodate these images. The space in the Management Module flash is too small to hold more than two images (primary and secondary) and hence, downloading a new image is not possible without deleting one of the images that is already present in the flash.

Before an image is copied onto the Management Module or Interface Module, the software now checks to refer to if there is enough space available in the Management Module flash to support the copy operation. If there is not enough free space available on the Management Module flash, the following error message will display on the user interface.

The 32 MB flash space is capable of holding two image. However, during the TFTP copy operation, it needs more buffer space. It is not possible to copy or update an existing image to a 32 MB flash, if there are two images in the flash already. If you try to copy or update an image, the following error message is displayed.

For TFTP copy operation, the following error message is displayed.

```
NetIron#copy tftp flash 10.20.10.62 mlxe04001b1.bin primary
There is not enough space on MP flash. Please clean up MP flash and retry, or use
"delete-first" option.
TFTP: Download to primary flash failed - Flash is full
```

For SCP copy operation, the following error message is displayed.

```
C:\>scp xm04001b1.bin lab@10.22.2.21:image:primary
There is not enough space on MP flash. Please clean up MP flash and retry, or use
"delete-first" option.
C:\>
```

In the example above the copy procedure is cancelled because there is not enough space on Management Module flash to copy the image. To make space for an image to be copied, you must clean up the flash space on the Management Module, and then retry copying the image again. You may also use the delete-first option, along with the CLI copy command, to make space for an image to be copied. The delete-first option allows you to delete existing target files on the Management Module flash.

The example below displays how the delete-first option is used. In this example, the existing secondary file image is removed from the flash to make space for a new image to be copied. The TFTP copy operation is able to successfully download the new image to the secondary flash.

```
NetIron#copy tftp flash 10.53.1.82 mlxe04001b1.bin secondary delete-first
Removing secondary from flash.
.....................................................................
........TFTP: Download to secondary flash done.
```

When the delete-first option is used, the existing target files are deleted only if there is enough free space to accommodate the copy operation. If, after the delete-first option is used and there is still a shortage of free space then the following error message will display.

```
NetIron#copy tftp flash 10.53.1.82 mlxe04001b1.bin secondary delete-first
There will not be enough space on MP flash even after deleting the target files.
Please clean up MP flash and retry.
```

Management focus

The **management focus** determines the default file system (flash memory or the flash card inserted in slot 1 or 2) to which a file management operation applies. When you power on or reload a PowerConnect system, by default, the management focus is on flash memory.

You can change the management focus from flash memory to a slot and subdirectory using the **cd** or **chdir** command. (For more information, refer to "Switching the management focus" on page 191.)

To determine the slot and subdirectory that have the current management focus, enter the **pwd** command. (For more information about this command, refer to "Determining the current management focus" on page 191.)

Most file management commands provide the option of specifying the file system to which the command applies. If you want the command to apply to the file system that has the current management focus, you do not need to specify the file system. If you want the operation to apply to the file system that does not have the current management focus, you must specify one of the following keywords:

- **flash** – indicates flash memory
- **slot1** – indicates the flash card inserted in slot 1
- **slot2** – indicates the flash card inserted in slot 2

For example, if you want to display a directory of files in flash memory and flash memory has the current management focus, you do not need to specify the **flash** keyword. However, if you want to display a directory of files for slot 1 and flash memory has the current focus, you must specify the **slot1** keyword.

Flash memory file system

The flash memory file system is flat, which means that it does not support subdirectories. As a result, you cannot create or delete subdirectories in this file system using the **md**/**mkdir** and **rd**/**rmdir** commands, respectively. Also, when specifying the syntax for the various file management commands, you will not need to specify a pathname to a subdirectory because it is not possible for a subdirectory to exist.

File naming conventions

A file name in the flash memory file system can contain a maximum of 31 characters. File name are case sensitive. The flash memory file system does not accept spaces as part of a file name.

The following characters are valid in file names:

- All upper and lowercase letters
- All digits
- Any of the following special characters:
  - $
  - %
  - '
  - -
  - _
  - @
  - ~
  - `
  - !
  - (
  -)
  - {
  - }
  - ^

- #
- &

PCMCIA flash card file system

The PCMCIA flash card file system is hierarchical, which means that it supports subdirectories. Therefore, you can create or delete subdirectories in this file system using the **md**/**mkdir** and **rd**/**rmdir** commands, respectively. Also, when specifying the syntax for the various file management commands, you may need to specify a pathname to a subdirectory as appropriate to manipulate a file in a subdirectory.

PCMCIA flash card subdirectories

The full path name for the location of a file can be a maximum of 256 characters. You can nest subdirectories as deep as you want as long as the full path name is 256 characters or less.

When you include a subdirectory path in a file management command, use a slash between each level. For example, to create a subdirectory for flash code and copy a flash image file to the subdirectory, enter commands such as the following.

```
NetIron# mkdir slot1 /switchCode/initial-release
```

These commands create two levels of subdirectories on the flash card in PCMCIA slot 1.

File and subdirectory naming conventions

The PCMCIA slots supports file names of up to 32 characters . File names are not case sensitive. Thus, the software considers the name “test.cfg” and “TEST.CFG” to be the same.

Files and subdirectory names can be up to 32 characters long, including spaces and the special characters listed. The following characters are valid in file and subdirectory names:

- All upper and lowercase letters
- All digits
- Spaces
- Any of the following special characters:
  - $
  - %
  - '
  - -
  - _
  - @
  - ~
  - `
  - !
  - (
  -)
  - {

- }
- ^
- #
- &

You can use spaces in a file or subdirectory name if you enclose the name in double quotes. For example, to specify a subdirectory name that contains spaces, enter a string such as the following: “a long subdirectory name”.

A subdirectory or file name can be a maximum of 256 characters long. A complete subdirectory path name cannot contain more than 256 characters.

There is no maximum file size. A file can be as large as the available flash card space.

Wildcards

Commands to display a directory of files, to change the read-write attribute of a file, or to delete files accept wildcards in the file name (<*file-name*>). With these commands, you can use “*” (asterisk) as a wildcard for any part of the name. For example, all the following values are valid for <*file-name*>:

- teststartup.cfg
- test*.cfg
- nmb02200.bin
- *.bin
- m*.bin
- m*.*

Formatting a flash card

The flash cards shipped with a management module are pre-formatted for the 16 FAT file system used by the modules. If you want to use a flash card that is not formatted for the 16 FAT file system, you need to reformat the flash card before you can store files on it.

CAUTION

Make sure the flash card is empty or does not contain files you want to keep. Formatting a flash card completely erases all files on the card.

CAUTION

Once you start the formatting process, you cannot stop it. Even if you enter CTRL-C to stop the CLI output and a new prompt appears, the formatting continues. Make sure you want to format the card before you enter the command.

To reformat a flash card in slot 2 on the management module, for example, enter the following command.

```
NetIron# format slot2
```

```
.................................................
.................................................
.................................................
.................................
 80809984 bytes total card space.
 80809984 bytes available on card.
     2048 bytes in each allocation unit.
    39458 allocation units available on card.
```

Syntax: format slot1 | slot2

The **slot1 | slot2** keyword specifies the PCMCIA slot that contains the flash card you are formatting.

Determining the current management focus

For conceptual information about management focus, refer to "Management focus" on page 187.

To determine which file system has the current management focus, enter the following command.

```
NetIron# pwd
Flash /flash/
```

In this example, the management focus is the flash memory.

In the following example, the management focus is the root directory of the flash card in slot 1.

```
NetIron# pwd
/slot1/
```

In the following example, the management focus is a subdirectory called "test" on the flash card in slot 1.

```
NetIron# pwd
/slot1/test/
```

Switching the management focus

The effect of file management commands depends on the file system that has the current management focus. For example, if you enter a command to delete a file and do not specify the location of the file, the software attempts to delete the file from the location that currently has the management focus.

By default, the management focus is on the flash memory on the management module. You can switch the focus from flash memory to flash cards in slot 1 or slot 2 on the management module using the **cd** or **chdir** commands, which have the same syntax and function exactly the same.

For example, to switch the focus from flash memory to the flash card in slot 2, enter the following command.

```
NetIron# cd /slot2
NetIron#
```

When you enter this command, the software changes the management focus to slot 2 then displays a new command prompt. If a slot you specify does not contain a flash card, the software displays the message shown in the following example.

```
NetIron# cd /slot2
Device not present
```

Syntax: **cd** <*directory-pathname*>

Syntax: chdir <*directory-pathname*>

For the <*directory-pathname*> parameter for both **cd** and **chdir** commands, specify /slot1 or /slot2 to switch the focus to slot 1 or slot 2, respectively. Specify /flash to switch the focus to flash memory.

After you have switched the focus to slot 2, you can specify the <*directory-pathname*> parameter to switch the focus to a subdirectory on a flash card inserted in slot 2. For example, to switch the focus from the root directory level (/) of slot 2 to the subdirectory named "PLOOK," enter the following command.

```
NetIron# cd /PLOOK
```

If you specify an invalid subdirectory path, the CLI displays a message such as the following.

```
NetIron# cd /PLOOK
Path not found
```

If you are certain the path you specified exists, make sure you are at the correct level to reach the path. For example, if you are already at the PLOOK level, the CLI cannot find the subdirectory "/PLOOK" because it is not a subdirectory from the level that currently has the management focus.

To change the management focus back to flash memory, enter the following command.

```
NetIron# cd /flash
NetIron#
```

Displaying a directory of the files

You can display a directory of the files in the flash memory on the management module, or on a flash card inserted in management module slot 1 or slot 2 using the **dir** or **ls** commands.

The software displays the directory of the file system that has the current management focus. By default, flash memory has the management focus. However, you do not need to change the focus to list the files on the file system that does not currently have management focus. In this case, you can specify the /<*path-name*>/ parameter with the **dir** or **ls** commands to display the directory of the desired file system.

For example, to display a directory of the files in flash memory, if flash memory has the management focus, enter the following command.

```
NetIron# dir
Directory of /flash/

07/28/2003 15:57:45              3,077,697 1060.tmp
07/28/2003 15:56:10              3,077,697 14082.tmp
07/28/2003 16:00:08              3,077,697 2084.tmp
07/25/2003 18:00:23                292,701 boot
00/00/0    00:00:00                     12 boot.ini
07/28/2003 14:40:19                840,007 lp-primary-0
07/28/2003 15:18:18                840,007 lp-secondary-0
07/28/2003 09:56:16                391,524 monitor
07/28/2003 15:08:12              3,077,697 primary
07/28/2003 16:02:23                  1,757 startup-config
07/25/2003 18:02:14                  1,178 startup.sj2
07/28/2003 14:28:47                  1,662 startup.spa
07/26/2003 12:16:29                  1,141 startup.vso
07/25/2003 18:11:01                  1,008 startup.vsr
07/28/2003 09:40:54                  1,554 startup.vsrp.ospf

               15 File(s)       14,683,339 bytes
                0 Dir(s)        15,990,784 bytes free
```

Syntax: dir | ls [<*path-name*>]

You can enter either **dir** or **ls** for the command name.

Specify the <*path-name*> parameter to display the following:

- The files that match the value for a flash memory directory, or flash card directory/subdirectory you specify
- The files that match the value for a name you specify

For example, to list only files that contain a .tmp suffix in flash memory, if flash memory is the current management focus, enter a command such as the following.

```
NetIron# dir *.tmp
Directory of /flash/
07/28/2003 15:57:45              3,077,697 1060.tmp
07/28/2003 15:56:10              3,077,697 14082.tmp
07/28/2003 16:00:08              3,077,697 2084.tmp

                3 File(s)        9,292,701 bytes
                0 Dir(s)        15,990,784 bytes free
```

For example, to display a directory of the files on the flash card in slot 2, if flash memory has the management focus, enter the following command.

```
NetIron# dir /slot2/
Directory of /slot2/

08/01/2003 18:25:28              3,092,508 PRIMARY
08/01/2003 18:28:06              3,092,508 primary.1234
08/01/2003 18:28:24                389,696 MONITOR
08/01/2003 18:28:30                389,696 MONITOR1
08/01/2003 18:28:01                389,696 MONITOR2
08/01/2003 18:28:03                389,696 MONITOR3
08/01/2003 18:29:04                389,696 MONITOR4
08/01/2003 18:29:12   <DIR>                DIR1
08/01/2003 18:32:03                389,696 1234567890.12345
08/01/2003 18:32:08                389,696 123456.123
08/01/2003 18:32:11                389,696 123456.123
08/01/2003 18:32:14                389,696 123456.123
08/01/2003 18:32:17                389,696 123456.123

               12 File(s)       10,081,976 bytes
                1 Dir(s)       114,577,408 bytes free
```

The following information is displayed for each file.

TABLE 38 CLI display of directory information

This field...	Displays...
File date	The date on which the file was placed in the flash memory or card, if the device system clock is set.
Time of day	The time of day at which the file was placed in the flash memory or card, if the device system clock is set.
File size	The number of bytes in the file.
Read-write attribute	If you have set the read-write attribute of the file to read-only, "R" appears before the file name. If the read-write attribute of the file is read-write (the default), no value appears in this column. For information, refer to "Changing the read-write attribute of a file" on page 198.
File name	The file name.
Long file name	This field applies to files on a flash card only. The longer file name applies if the file was created on a PC and the name is longer than the 8.3 format.

The directory also lists the total number of files that match the parameters you specified, the total number of bytes used by all the files, and the number of bytes still free.

Displaying the contents of a file

You can display the contents of a file in the flash memory on the management module or on a flash card inserted in management module slot 1 or slot 2.

The software displays the specified file in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to display the file in a file system that does not currently have management focus. In this case, you can specify the */<directory>/ <path-name>* parameter with the **more** command to display the file in the desired file system.

For example, to display the contents of a file in flash memory, if flash memory has the current management focus, enter a command such as the following.

```
NetIron# more cfg.cfg
```

Syntax: more [/<*directory*>/]<*file-name*>

Use the <*directory*> parameter to specify a directory in a file system that does not have current management focus.

Use the <*path-name*> parameter to specify the file you want to display.

For example, to display the contents of a file on the flash card in slot 2, if flash memory has the current management focus, enter a command such as the following.

```
NetIron# more /slot2/cfg.cfg
```

Displaying the hexadecimal output of a file

You can display the hexadecimal output of a file in flash memory on the management module or on a flash card inserted in management module slot 1 or slot 2.

The software displays the hexadecimal output of a specified file in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to display the hexadecimal output of a file in a file system that does not currently have management focus. In this case, you can specify the /<*directory*>/<*file-name*> parameter with the **hd** command to display the output of the file in the desired file system.

For example, to display the hexadecimal output of a file in flash memory, if flash memory has the current management focus, enter the following command.

```
NetIron# hd cfg.cfg
```

Syntax: [no] hd [/<*directory*>/]<*file-name*>

Use the <*directory*> parameter to specify a directory in a file system that does not have current management focus.

Use the <*file-name*> parameter to specify a file for which you want to display the hexadecimal output.

For example, to display the hexadecimal output of a file in a flash card inserted in slot 2, if flash memory has the current management focus, enter the following command.

```
NetIron# hd /slot2/cfg.cfg
```

Creating a subdirectory

Create a subdirectory in the flash card file system using the **md** and **mkdir** commands, which have the same syntax and function exactly the same.

NOTE
You cannot create subdirectories in the flash memory file system. Therefore, the **md** and **mkdir** commands do not apply to the flash memory file system.

The software creates a subdirectory in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to create a subdirectory in a file system that does not currently have management focus. In this case, you can specify the **slot1** or **slot2** keyword with the **md** or **mkdir** command to create the subdirectory in the desired file system.

For example, to create a subdirectory on the flash card inserted in slot 2, if the flash memory has current management focus, enter a command such as the following.

```
NetIron# mkdir slot2 TEST
```

Syntax: [no] md | mkdir [slot1 | slot2] <*dir-name*>

You can enter either **md** or **mkdir** for the command name.

Specify the **slot1** or **slot2** keyword to create a subdirectory on the flash card in slot 1 or slot 2, respectively. If you do not specify one of these parameters, the command applies to the file system that currently has the management focus.

The <*dir-name*> parameter specifies the subdirectory name. You can enter a name that contains any combination of the following characters. Do not enter a slash " / " in front of the name. Remember, a file name preceded by a slash represents the absolute path name (/flash, /slot1, or /slot2).

- All upper and lowercase letters
- All digits
- Spaces
- Any of the following special characters:
  - $
  - %
  - '
  - -
  - _
  - @
  - ~
  - `
  - !
  - (
  -)
  - {
  - }
  - ^
  - #
  - &

You can use spaces in a subdirectory name if you enclose the name in double quotes. For example, to specify a subdirectory name that contains spaces, enter a string such as the following: "a long subdirectory name".

A subdirectory name can be a maximum of 256 characters long. A complete subdirectory path name cannot contain more than 260 characters.

The name is not case sensitive. You can enter upper- or lowercase letters, however the CLI displays the name using uppercase letters.

To verify successful creation of the subdirectory, enter a command such as the following to change to the new subdirectory level.

```
NetIron# chdir /slot2/TEST
Current directory of slot2 is: /TEST
```

For information about changing the directory using the **cd** and **chdir** commands, refer to "Switching the management focus" on page 191.

Removing a subdirectory

You can remove a subdirectory from the flash card file system using the **rd** and **rmdir** commands, which have the same syntax and function exactly the same.

NOTE
You cannot remove subdirectories from the flash memory file system. Therefore, the **rd** and **rmdir** commands do not apply to the flash memory file system.

NOTE
You can remove a subdirectory only if the subdirectory does not contain files or other subdirectories.

The software will remove a subdirectory from the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to remove a subdirectory from a file system that does not currently have management focus. In this case, you can specify the **slot1** or **slot2** keyword with the **rd** or **rmdir** command to remove the subdirectory from the desired file system.

For example, to remove a subdirectory from the flash card inserted in slot 2, if the flash memory has current management focus, enter a command such as the following.

```
NetIron# rmdir slot2 TEST
```

Syntax: [no] rd | rmdir [slot1 | slot2] <*dir-name*>

You can enter either **rd** or **rmdir** for the command name.

Specify the **slot1** or **slot2** keyword to remove a subdirectory on the flash card in slot 1 or slot 2, respectively. If you do not specify one of these parameters, the command applies to the file system that currently has the management focus.

The <*dir-name*> parameter specifies the subdirectory you want to delete. You can enter a path name if the subdirectory is not in the current directory.

If you receive a message such as the following, enter the **pwd** command to verify that the management focus is at the appropriate level of the directory tree.

```
NetIron# rmdir TEST
rmdir /slot1/test/dir1/temp failed - File not found
```

For information about using the **pwd** command, refer to "Determining the current management focus" on page 191.

Renaming a file

You can rename a file in the flash memory on the management module or on a flash card inserted in management module slot 1 or slot 2 using the **rename** or **mv** command.

The software renames the file in the file system that has the current management focus flash memory by default). However, you do not need to change the focus to rename the file in a file system that does not currently have management focus. In this case, you can specify the */<directory>/<old-file-name>* /<directory>/*<new-file-name>* parameter with the **rename** or **mv** command to rename the file in the desired file system.

For example, to rename a file in flash memory, if flash memory has the current management focus, enter a command such as the following.

```
NetIron# rename oldname newname
```

If the command is successful, the CLI displays a new command prompt.

Syntax: [no] rename | mv *[/<directory>/]<old-file-name> [/<directory>/]<new-file-name>*

You can enter either **rename** or **mv** for the command name.

The */<directory>/* parameter specifies a directory in a file system that does not have current management focus.

The *<old-file-name>* parameter specifies the original filename that you want to change.

The *<new-file-name>* parameter specifies the new filename that you want to assign to the original file.

For example, to rename a file on the flash card inserted in slot 2, if flash memory has the current management focus, enter a command similar to the following.

```
NetIron# rename /slot2/oldname /slot2/newname
```

Changing the read-write attribute of a file

You can specify the read-write attribute of a file on a flash card as follows:

- **Read-only** – You can display or copy the file but you cannot replace (copy over) or delete the file.
- **Read-write** – You can replace (copy over) or delete the file. This is the default.

NOTE
All files in flash memory are set to the read-write attribute, which cannot be changed. You cannot change this attribute. Therefore, the **attrib** command does not apply to the flash memory file system.

To determine the current setting of the read-write attribute for a file, use the **dir** command to list the directory information for the file. Files set to read-only are listed with "R" in front of the file name. For information about the **dir** command, refer to "Displaying a directory of the files" on page 192.

The software will change the read-write attribute of the file in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to change this file attribute in a file system that does not currently have management focus. In this case, you can specify the **slot1** or **slot2** keyword with the **attrib** command to change the attribute of the file in the desired file system.

For example, to change the attribute of a file in slot2 to read-only, if flash memory has the management focus, enter a command similar to the following.

```
NetIron# attrib slot2 ro goodcfg.cfg
```

Syntax: [no] attrib [slot1 | slot2] ro | rw *<file-name>*

Specify the **slot1** or **slot2** keyword to change the attribute of a file on the flash card in slot 1 or slot 2, respectively. If you do not specify one of these keywords, the command applies to the file system that currently has the management focus.

The **ro** parameter specifies that the attribute of the file is set to read-only. The **rw** parameter specifies that the attribute of the file is set to read-write.

The *<file-name>* parameter specifies the file for which to change the attribute.

For example, to change the attribute of all files on the flash card in slot 2 to read-only, if flash memory has the current management focus, enter a command similar to the following.

```
NetIron# attrib slot2 ro *.*
```

Deleting a file

You can delete a file from flash memory or a flash card inserted in slot 1 or slot 2 on the management module using the **delete** or **rm** command.

NOTE
The **delete** or **rm** command deletes all files in a file system unless you explicitly specify the files you want to delete.

NOTE
The software does not support an undelete option for the flash memory file system. Be sure you really want to delete the file before you execute this command.

The software will delete the file in the file system that has the current management focus. By default, flash memory has the management focus. However, you do not need to change the focus to delete the file in a file system that does not currently have management focus. In this case, you can specify the */<directory>/<file-name>* parameter with the **delete** or **rm** command to delete the file in the desired file system.

For example, to delete a file in flash memory, if flash memory has the current management focus, enter a command similar to the following.

```
NetIron# delete cfg.cfg
```

If the command is successful, the CLI displays a new command prompt.

Syntax: delete | rm [slot1 | slot2] [*<directory>] [<file-name>*]

You can enter either **delete** or **rm** for the command name.

Specify the **slot1** or **slot2** keywords to delete all files on the flash card in slot 1 or slot 2, respectively.

The *<directory>* parameter specifies the directory in a file system that does not have the current management focus.

The *<file-name>* parameter specifies the file that you want to delete.

For example, to delete all files with names that start with “test” from flash memory, if flash memory has the current management focus, enter a command similar to the following.

```
NetIron# delete test*.*
```

For example, to delete all files on the flash card in slot 2, if flash memory has the current management focus, you can enter one of the following commands.

```
NetIron# delete /slot2/
```

or

```
NetIron# delete slot2
```

Recovering ("undeleting") a file

You can recover or undelete a file you have deleted from a flash card file system using the **undelete** command.

NOTE
You can not recover or undelete a file from the flash memory file system. Therefore, the **undelete** command does not apply to the flash memory file system.

The software will recover the file in the file system that has the current management focus (flash memory by default). If you want to recover a file in a file system that does not have the current management focus, you must switch the management focus to the desired file system using the **cd** command. For more information about switching the management focus, refer to "Switching the management focus" on page 191.

For example, to undelete a file on the flash card in slot 2, if flash memory has the current management focus, enter a command such as the following.

```
NetIron# cd slot2
NetIron# undelete
Undelete file ?RIMARY ? (enter y or n) :y
Input one character: P
File recovered successfully and named to PRIMARY
```

For each file that can be undeleted from the flash card in slot 2, the CLI displays the remaining name entry in the file directory and prompts you for the first character of the file name. You can enter any valid file name character. You do not need to enter the character that was used before in the deleted file name.

Once you enter a character and the CLI undeletes the file, the CLI continues with the next file that can be undeleted. For each file, specify "y" or "n", and specify a first character for the files that you select to undelete.

NOTE
When you delete a file from a flash card, the CLI leaves the file intact but removes the first letter in the file name from the file directory. However, if you save file changes or new files that use part of the space occupied by the deleted file, you cannot undelete the file. The **undelete** command lists only the files that can be undeleted.

To end the undelete process, enter CTRL + C.

Appending a file to another file

You can append a file in flash memory or on a flash card to the end of another file in one of these file systems.

The software will append one file to another in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to append one file to another in a file system that does not currently have management focus. In this case, you can specify the /<*source-dir-path*>/ or /<*dest-dir-path*>/ parameters with the **append** command to append one file to another in the desired file system.

To append one file to another in flash memory, if flash memory has the current management focus, enter a command similar to the following.

```
NetIron# append newacls.cfg startup-config.cfg
```

Syntax: [no] **append** [<*source-file-system*> <*dest-file-system*>] [/<*source-dir-path*>/]<*source-file-name*> [/<*dest-dir-path*>/]<*dest-file-name*>

Specify the <*source-file-system*> and <*dest-file-system*> parameters when you are appending a file on one file system to a file on another file system.

The [/<*source-dir-path*>/] <*source-file-name*> parameter specifies the file you are appending to the end of another file. If the file is not located in the current subdirectory (the subdirectory that currently has the management focus), specify the subdirectory path in front of the file name.

The [/<*dest-dir-path*>/]<*dest-file-name*> parameter specifies the file to which you are appending the other file. If the file is not located in the current subdirectory, specify the subdirectory path in front of the file name.

For example, to append a file in the root directory of slot 1 to another file in a subdirectory of slot 2, enter a command similar to the following.

```
NetIron# append slot1 slot2 newacls.cfg /TEST/startup-config.cfg
```

Copying files using the copy command

For information about copying files using the **copy** command while upgrading software images, refer to "Basic Tasks in the Software Upgrade Process" in the *PowerConnect B-MLXe Hardware Installation Guide.*

You can perform the following additional copy operations using the **copy** command:

- Copy files from one flash card to the other
- Copy files between a flash card and the flash memory on the management module
- Copy software images between active and standby management modules
- Copy files from a management module to an interface module
- Copy management module PowerConnect Multi-Service IronWare images from flash memory to a TFTP server
- Copy files between a flash card and a TFTP server
- Copy a startup-config file between a flash card and flash memory on the management module
- Copy a startup-config file between flash memory on the management module and a TFTP server
- Copy the running-config to a flash card or a TFTP server
- Load a running-config from a flash card or TFTP server into the running-config (loading ACLs only) on the device

NOTE
Since the copy options require you to explicitly specify the flash card, you can perform a copy regardless of the flash card that currently has the management focus.

Copying files from one flash card to the other

To copy a file from one flash card to the other, enter the following command.

```
NetIron# copy slot1 slot2 sales.cfg
```

Syntax: copy <*from-card*> <*to-card*> [/<*from-dir-path*>/]<*from-name*> [/<*to-dir-path*>/][<*to-name*>]

For the <*from-card*> and <*to-card*> parameters, you can specify **slot1** or **slot2**.

The command shown in the example copies a file from the flash card in slot 1 to the flash card in slot 2. In this case, the software uses the same name for the original file and for the copy. Optionally, you can specify a different file name for the copy.

Copying files between a flash card and flash memory

To copy a file from a flash card to the primary area in flash memory, enter a command similar to the following.

```
NetIron# copy slot1 flash nmpr02200.bin primary
```

Syntax: copy slot1 | slot2 flash [/<*from-dir-path*>/]<*from-name*> **monitor | primary | secondary**

To copy a file from flash memory to a flash card, enter a command similar to the following.

```
NetIron# copy flash slot2 nmpr02200.bin primary
```

Syntax: copy flash slot1 | slot2 <*source-name*> **monitor | primary | secondary | startup-config** [<*dest-name*>]

The command in this example copies a PowerConnect Multi-Service IronWare image file from the primary area in flash memory onto the flash card in slot 2. In this case, the software uses the same name for the source file and for the destination file. Optionally, you can specify a different file name for the destination file.

Copying software images between active and standby management modules

To copy the monitor image from flash memory of the active management module to flash memory of the standby module, enter the following command.

```
NetIron# copy flash flash monitor standby
```

To copy the PowerConnect Multi-Service IronWare image from the secondary location in flash memory on the active management module to the primary location in flash memory, enter the following command.

```
NetIron# copy flash flash primary
```

Syntax: copy flash flash primary [standby]

Specify the optional **standby** keyword to copy the PowerConnect Multi-Service IronWare image from the secondary location in flash memory on the active management module to the primary location in flash memory on the standby module.

To copy the PowerConnect Multi-Service IronWare image from the primary location in flash memory on the active management module to the secondary location in flash memory on the active module, enter the following command.

```
NetIron# copy flash flash secondary
```

Syntax: copy flash flash secondary [standby]

Specify the optional **standby** keyword to copy the PowerConnect Multi-Service IronWare image from the primary location in the flash memory on the active management module to the secondary location in the flash memory on the standby module.

Copying files from a management module to an interface module

You can copy a software image or other type of file from flash memory on the management module to the flash memory on one or all interface modules.

For example, to copy the monitor image on the interface module from the management module to all interface modules, enter a command similar to the following.

```
NetIron# copy flash lp nlb02200.bin monitor all
```

Syntax: copy flash lp <*source-file*> **monitor | primary | secondary** <*slot-number*> **| all**

For example, to copy a file called test.cfg from the management module to the interface module in chassis slot 1, enter a command similar to the following.

```
NetIron# copy flash lp test.cfg lptest.cfg 1
```

Syntax: copy flash lp <*source-file*> <*dest-file*> <*slot-number*> **| all**

Copying PowerConnect Multi-Service IronWare images from flash memory to a TFTP Server

You can copy PowerConnect Multi-Service IronWare images from the primary and secondary locations in flash memory on the management module to a TFTP server.

For example, to copy the PowerConnect Multi-Service IronWare image in the secondary location in flash memory to a TFTP server, enter a command similar to the following.

```
NetIron# copy flash tftp 10.10.10.1 secondary.bak secondary
```

Syntax: copy flash tftp <*ip-addr*> <*dest-file-name*> **primary | secondary**

Copying files between a flash card and a TFTP server

Use the following methods to copy files between a flash card and a TFTP server.

NOTE
The PowerConnect system must have network access to the TFTP server.

To copy a file from a flash card to a TFTP server, enter a command similar to the following.

```
NetIron# copy slot1 tftp 192.168.1.17 notes.txt
```

Syntax: copy slot1 | slot2 tftp <*ip-addr*> [/<*from-dir-path*>/]<*source-file*> [<*dest-file*>]

The command in this example copies a file from slot 1 to a TFTP server. In this case, the software uses the same name for the source file and for the destination file. Optionally, you can specify a different file name for the destination file.

To copy a software image from a TFTP server to a flash card, enter a command similar to the following.

```
NetIron# copy tftp slot1 192.168.1.17 nmpr02200.bin primary
```

Syntax: copy tftp slot1 | slot2 <*ip-addr*> [/<*from-dir-path*>/]<*source-file*> <*path-name*> **| monitor | primary | secondary**

The command in this example copies the primary PowerConnect Multi-Service IronWare image from a TFTP server to a flash card in slot 1.

Copying the startup-config file between a flash card and flash memory

Use the following methods to copy a startup-config file between flash memory and a flash card. By default, the PowerConnect router uses the startup-config in the primary area of flash memory when you boot or reload the device.

NOTE
The PowerConnect router cannot configure from a startup-config file on a flash card. You cannot boot or reload from a flash card.

To copy a startup-config file from a flash card to flash memory, enter a command similar to the following.

```
NetIron# copy slot1 startup-config test2.cfg
```

Syntax: copy slot1 | slot2 startup-config [/<*from-dir-path*>/]<*file-name*>

This command copies a startup configuration named test2.cfg from the flash card in slot 1 into the flash memory on the device. The next time you reboot or reload, the device uses the configuration information in test2.cfg.

To copy the startup-config file on the device from flash memory onto a flash card, enter a command similar to the following.

```
NetIron# copy startup-config slot1 mfgtest.cfg
```

Syntax: copy startup-config slot1 | slot2 [/<*to-dir-path*>/]<*to-name*>

This command copies the startup configuration from the flash memory on the device to a flash card in slot 1 and names the file mfgtest.cfg.

Copying the startup-config file between flash memory and a TFTP server

Use the following methods to copy a startup-config between flash memory and a TFTP server to which the PowerConnect system has access. By default, the device configures from the startup-config in the primary area of flash memory when you boot or reload the device.

To copy the startup-config on the device from flash memory to a TFTP server, enter a command similar to the following.

```
NetIron# copy startup-config tftp 10.10.10.1 /backups/startup.cfg
```

Syntax: copy startup-config tftp <*ip-addr*> [/<*to-dir-path*>]<*to-name*>

To copy a startup-config file from a TFTP server to flash memory, enter a command similar to the following.

```
NetIron# copy tftp startup-config 10.10.10.1 test.cfg
```

Syntax: copy tftp startup-config <*ip-addr*> [/<*from-dir-path*>]<*from-name*>

Copying the running-config to a flash card or a TFTP server

Use the following method to copy the config file on the PowerConnect router to a flash card or a TFTP server. The running-config contains currently active configuration information for the device. When you copy the running-config to a flash card or TFTP server, you are making a copy of the current configuration, including any configuration changes you have not saved to the startup-config.

To copy the running configuration for the device into a file on a flash card, enter a command similar to the following.

```
NetIron# copy running-config slot1 runip.1
```

Syntax: copy running-config slot1 | slot2 [/<*to-dir-path*>/]<*to-name*>

To copy the running configuration for the device into a file on a TFTP server, enter a command such as the following.

```
NetIron# copy running-config tftp 10.10.10.1 runip.1
```

Loading a running-config from a flash card or a TFTP server

Use the following method to load configuration commands into the active configuration for the PowerConnect router.

NOTE
A configuration file that you create must follow the same syntax rules as the startup-config the device creates. Refer to "Dynamic Configuration Loading" in the *PowerConnect B-MLXe Hardware Installation Guide.*

To copy a running-config from a flash card, enter a command such as the following.

```
NetIron# copy slot2 running-config runacl.2
```

Syntax: copy slot1 | slot2 running-config [/<*from-dir-path*>/]<*from-name*>

The command in this example changes the active configuration for the device based on the information in the file.

To copy a running-config from a TFTP server, enter a command similar to the following.

```
NetIron# copy tftp running-config 10.10.10.1 run.cfg overwrite
```

Syntax: copy tftp running-config <*ip-addr*> [/<*from-dir-path*>/]<*from-name*> **[overwrite]**

This command copies a running-config from a TFTP server and overwrites the active configuration for the device.

NOTE
You cannot use the overwrite option from non-console sessions, as it will disconnect the session.

Copying files using the cp command

Use the **cp** command to do the following:

- Copy files from flash memory to flash memory
- Copy files from flash memory to a flash card or vice versa
- Copy files from one flash card to another flash card

The software will copy a file in a file system to another location in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to copy a file from one location to another in a file system that does not currently have management focus. In this case, you can specify the /<*source-dir-path*>/ or /<*dest-dir-path*>/ parameters with the **cp** command to copy a file to or from a file system that does not have current management focus.

For example, to copy a file from flash memory, which has the current management focus, to flash memory, enter a command similar to the following.

```
NetIron# cp primary primary2
```

For example, to copy a file from flash memory, which has the current management focus, to the flash card in slot 2, enter a command similar to the following.

```
NetIron# cp new.cfg /slot2/cfg/new.cfg
```

Syntax: **cp** [<*source-dir-path*>]<*source-file-name*> [<*dest-dir-path*>]<*dest-file-name*>

The <*source-dir-path*> parameter specifies the directory pathname of the source file. Specify this parameter if the source file is in a file system that does not have current management focus. The <*source-file-name*> specifies the name of the file you want to copy.

The <*dest-dir-path*> parameter specifies the directory pathname of the destination file. Specify this parameter if you want to copy the source file to a file system that does not have current management focus. The <*dest-file-name*> specifies the name of the file you copied to a new destination.

For example, to copy a file from a flash card in slot 2 to flash memory, which has current management focus, enter the following command.

```
NetIron# cp /slot2/cfg/new.cfg new.cfg
```

For example, to copy a file from a flash card in slot 1 to a flash card in slot 2, neither of which has current management focus, enter the following command.

```
NetIron# cp /slot1/cfg/new.cfg /slot2/cfg/new.cfg
```

Loading the software

By default, the management module loads an PowerConnect Multi-Service IronWare image from the primary location in flash memory. You can change the PowerConnect Multi-Service IronWare image source for the system to one of the following sources for a single reboot or for all future reboots:

- The secondary location in flash memory
- A flash card inserted in slot 1 or 2
- A TFTP server
- A BOOTP server

If you specify a source other than the primary location in flash memory and for some reason the source or the PowerConnect Multi-Service IronWare image is unavailable, the system uses the primary location in flash memory as a default backup source.

Rebooting from the system

To use a source besides the IronWare image in the primary location in flash memory for a single reboot, enter a command similar to the following at the Privileged EXEC level of the CLI.

```
NetIron# boot system slot1 /slot1/mlxe03000.bin
```

The command in this example reboots the system using the image mlxe03000.bin located on the flash card in slot 1. This example assumes that the flash card in slot 1 is not the management focus.

Syntax: boot system slot1 | **slot2** [/<*dir-path*>/]<*file-name*>

The **slot1** | **slot2** keywords specify the flash card slot.

The <*file-name*> parameter specifies the file name. If the file is in a subdirectory, specify the subdirectory path in front of the file name. If the file name you specify is not a full path name, the CLI assumes that the name (and path, if applicable) you enter are relative to the subdirectory that currently has the management focus.

NOTE
This command also is supported at the boot PROM.

For example, to reboot the system using the image mlxe03000.bin on a TFTP server, enter a command similar to the following.

```
NetIron# boot system tftp 10.10.10.1 mlxe03000.bin
```

Syntax: boot system tftp <*ip-address*> <*file-name*>

The <*ip-address*> parameter specifies the address of the TFTP server on which the desired image resides.

The <*file-name*> parameter specifies the name of the PowerConnect Multi-Service IronWare image on the TFTP server.

For example, to reboot the system using the secondary location in flash memory, enter the following command.

```
NetIron# boot system flash secondary
NetIron# Are you sure? (enter 'y' or 'n'): y
```

Syntax: boot system flash secondary

To reboot the system from a BOOTP server, enter the following command.

```
NetIron# boot system bootp
```

Syntax: boot system bootp

Configuring the boot source for future reboots

To change the PowerConnect Multi-Service IronWare image source from the primary location in flash memory to another source for future reboots, enter a command similar to the following at the global CONFIG level of the CLI.

```
NetIron(config)# boot system slot1 mlxe03000.bin
```

The command in this example sets PCMCIA slot 1 as the primary boot source for the PowerConnect router. When you reload the software or power cycle the device, the device will look for the PowerConnect Multi-Service IronWare image on the flash card in slot 1.

Syntax: **boot system slot1** <*file-name*> | **slot2** <*file-name*> | **flash secondary** | **tftp** <*ip-address*> <*file-name*> | **bootp**

NOTE
The command syntax is the same for immediately reloading and for changing the primary source, except the <*file-name*> must be the full path name. You cannot specify a relative path name. If the first character in the path name is not a slash (/), the CLI treats the name you specify as relative to the root directory.

How the device responds to the command depends on whether you enter the command at the Privileged EXEC level or the global CONFIG level.

If you enter multiple **boot system** commands at the global CONFIG level, the software places them in the running-config in the order you enter them, and saves them to the startup-config in the same order when you save the configuration. When you reload or power cycle the device, the device tries the boot sources in the order they appear in the startup-config and running-config.

Saving configuration changes

You can configure the PowerConnect system to save configuration changes to a startup-config in flash memory or on a flash card in slot 1 or 2.

Displaying the current location for saving configuration changes

Enter the following command at the Privileged EXEC level of the CLI to display the current save location for the startup-config.

```
NetIron# locate startup-config
Startup-config data location is flash memory
```

Specifying the location for saving configuration changes

By default, when you save configuration changes, the changes are saved to the startup-config in flash memory.
To change the save location to a flash card in slot 1 or 2, enter a command similar to the following.

```
NetIron# locate startup-config slot1 router1.cfg
NetIron# write memory
```

The first command in this example sets the device to save configuration changes to the file named “switch1.cfg” in the flash card in slot 1. The second command saves the running-config to the router1.cfg file on the flash card in slot 1.

NOTE
In this example, after you save the configuration changes using the **write memory** command, the router1.cfg file will include the command that designates slot 1 as the save location for configuration changes.

Syntax: **locate startup-config [slot1 | slot2 | flash-memory] [/**<*dir-path-name*>**/]**<*file-name*>

The **locate** command is used only for saving the startup-config file to a different location. But once after reload, the system always picks up the startup-config file from the flash memory.

The **slot1**, **slot2**, and **flash-memory** keywords specify the flash card in slot 1 or slot 2 or flash memory as the save location for configuration changes.

Specify the <*dir-path-name*> parameter if you want to save the configuration changes to a directory other than the root directory of a flash card file system.

The <*file-name*> parameter indicates the name of the saved configuration file.

To change the save location back to flash memory, enter a command similar to the following.

```
NetIron# locate startup-config flash-memory router1.cfg
NetIron# write memory
```

File management messages

The following table lists the messages the CLI can display in response to file management commands.

TABLE 39 Flash card file management messages

This message...	Means...
File not found	You specified a file name that the software could not find. Verify the command you entered to make sure it matches the source and destination you intended for the file operation.
Current directory is: <*dir-path*>	You have successfully changed the management focus to the slot and subdirectory indicated by the message.
Path not found	You specified an invalid path.
There is not enough space on the card	The flash card does not have enough space to hold the file you are trying to copy to it.
Access is denied	You tried to copy or delete a file that has the read-only attribute.
A duplicate file name exists	You tried to rename a file using a name that is already in use by another file.
Fatal error, can not read or write media	A hardware error has occurred. One possible cause of this message is removing the flash card while a file operation involving the card was in progress.
There is sharing conflict between format command and other read/write operations	The flash card is currently undergoing formatting. This message also appears if you enter a command to format the card while the card is being accessed for another file operation.
Invalid DOS file name	A filename you entered contains an invalid character (for example, ":" or "\").
File recovered successfully and named <*file-name*>	A file you tried to recover was successfully recovered under the name indicated in the message

6 Verifying available flash space on the management module before an image is copied

NetIron MLX Link Aggregation 7

Overview

This chapter describes how to configure Link Aggregation Groups (LAG) for the NetIron MLX. You can use a single interface to configure any of the following LAG types:

- **Static LAGs** – These LAG groups are manually-configured aggregate links containing multiple ports.
- **Dynamic LAGs** – This LAG type uses the Link Aggregation Control Protocol (LACP), to maintain aggregate links over multiple port. LACP PDUs are exchanged between ports on each router to determine if the connection is still active. The LAG then shuts down ports whose connection is no longer active.
- **Keep Alive LAGs** – In a Keep Alive LAG a single connection between a single port on 2 PowerConnect routers is established. In a keep alive LAG, LACP PDUs are exchanged between the 2 ports to determine if the connection between the routers is still active. If it is determined that the connection is no longer active, the ports are blocked.

LAG formation rules

The LAG formation rules are mentioned below:

- You cannot configure a port concurrently as a member of a static, dynamic, or keep-alive LAG.
- Any number or combination of ports between 1 and 32 within the same chassis can be used to configure a LAG. The maximum number of LAG ports is checked when adding ports to a LAG.
- All ports configured in a LAG must be of equal bandwidth. For example all 10 G ports.
- All ports configured in a LAG must be configured with the same port attributes.
- LAG formation rules are checked when a static or dynamic LAG is deployed.
- A LAG must have its primary port selected before it can be deployed.
- All ports configured in a LAG must be configured in the same VLAN.
- All ports must have the same PBR configuration before deployment. During deployment, the configuration on the primary port is replicated to all ports. On undeployment, each port inherits the same PBR configuration.
- VLAN and inner-VLAN translation

  The LAG is rejected if any LAG port has VLAN or inner-VLAN translation configured
- Layer 2 requirements:

  The LAG is rejected if the LAG ports:

  - Do not have the same untagged VLAN component.
  - Do not share the same SuperSpan customer ID (CID).

  - Do not share the same VLAN membership or do not share the same uplink VLAN membership
  - Do not share the same protocol-VLAN configuration
  - Are configured as mainly primary and secondary interfaces
- Layer 3 requirements:

  The LAG is rejected if any of the secondary LAG port has any Layer 3 configurations, such as IPv4 or IPv6 address, OSPF, RIP, RIPNG, IS-IS, and so on.
- Layer 4 (ACL) requirements:
  - All LAG ports must have the same ACL configurations; otherwise, the LAG is rejected.
  - A LAG cannot be deployed if any of the member ports has ACL-based mirroring configured on it.
  - A port with ACL-based mirroring configured on it cannot be added to a LAG.
- The router can support up to 256 LAGs, and each LAG can contain up to 32 member ports. If the router is configured to support 64 LAGs by using the **system-max trunk-num** command, the maximum number of LAG ports is 32. If the system-max trunk-num is set to 256, the maximum number of LAG ports supported is 8. The default system-max trunk-num is set to 128, and each LAG can have up to 16 member ports.
- Ports can be in only one LAG group. All the ports in a LAG group must be connected to the same device at the other end. For example, if port 1/4 and 1/5 in Device 1 are in the same LAG group, both ports must be connected to ports in Device 2 or in Device 3. You cannot have one port connected to Device 2 and another port connected to Device 3.
- All LAG member properties must match the primary port of the LAG with respect to the following parameters:
  - Port tag type (untagged or tagged port)
  - Port speed and duplex
  - TOS-based Configuration – All ports in the LAG must have the same TOS-based QoS configuration before LAG deployment, During deployment the configuration on the primary port is replicated to all ports and on undeployment, each port inherits the same TOS-based QoS configuration.

  To change port parameters, you must change them on the primary port. The software automatically applies the changes to the other ports in the LAG.
- The router can support the following LAG/member port configurations:
  - 256 LAGs with each containing 8 member ports.
  - 128 LAGs with each containing 16 member ports.
  - 64 LAGs with each containing 32 member ports.
  - 32 LAGs with each containing 64 member ports.

  You can change the number of LAGs and member ports by using the **system-max trunk-num** *num* command. The valid values are 32, 64, 128, and 256. By default, the router is configured to support 128 LAGs with each containing 16 member ports.
- Make sure the device on the other end of the LAG link can support the same number of ports in the link.

Figure 3 displays an example of a valid, Keep ALIVE LAG link between two devices. This configuration does not aggregate ports but uses the LACP PDUs to maintain the connection status between the two ports.

FIGURE 3 Example of a 1-port keep alive LAG

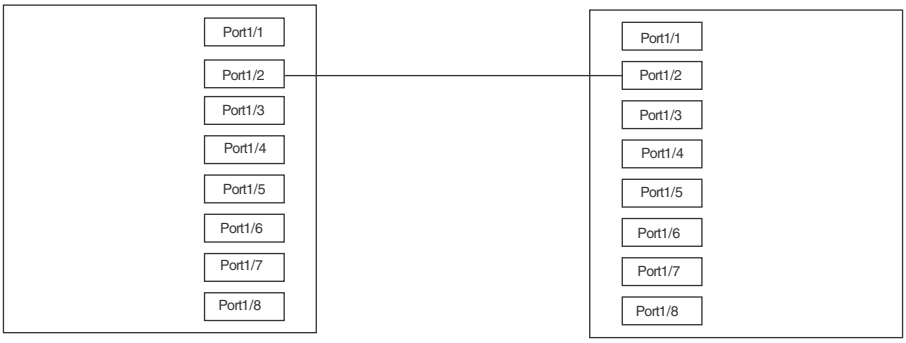


Figure 4 shows an example of a valid 2-port LAG link between devices where the ports on each end are on the same interface module. Ports in a valid 2-port LAG on one device are connected to two ports in a valid 2-port LAG on another device.

FIGURE 4 Example of 2-port LAG

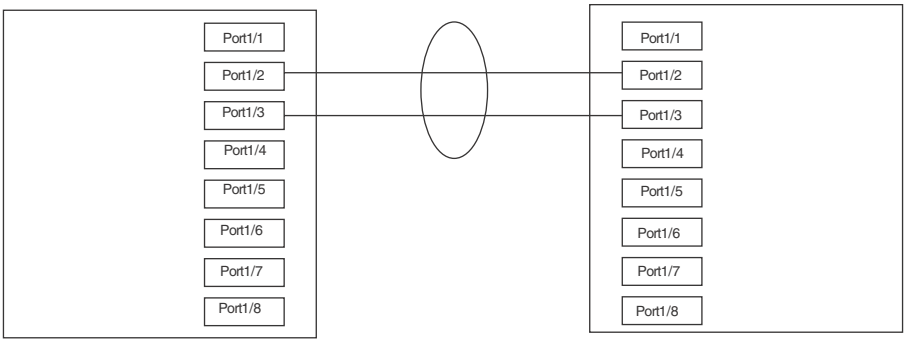
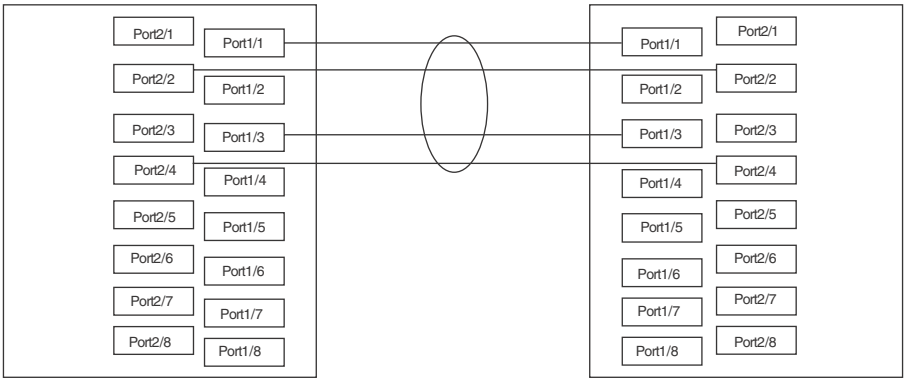


Figure 5 shows an example of two devices connected over a 4 port LAG where the ports on each end of the LAG are on different interface modules.

FIGURE 5 Examples of multi-slot, multi-port LAG



LAG load sharing

PowerConnect devices can be configured for load sharing over a LAG by either of the following methods:

- Hash Based Load Sharing
- Per Packet Load Sharing

Each of these methods, that are described in the following sections, are configured per LAG using the **trunk-type** command as described in "Configuring load sharing type" on page 220.

Hash based load sharing

The PowerConnect router shares the traffic load evenly across the ports in LAG group, while ensuring that packets in the flow are not reordered. Individual flows are assigned a LAG index to identify them. Hash based load sharing algorithm was introduced with the following enhancements:

- Better Distribution
- Support for 32-port LAGs
- An increased number of fields in the packet header that can be used for load balancing
- Enhanced load sharing in configurations of ECMP with LAGs.

Traffic from each flow is then distributed across the ports in the LAG group using a hash index as follows:

- For L2 switching, the hash index is based on the following:
  - IPv4 or IPv6 traffic: source MAC address and destination MAC address, IPv4v6 source and destination address, VLAN-ID, IPv4 protocol number or IPv6 next header and TCP or UDP source port and TCP or UDP destination port.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:
– Packet is non-fragmented and without option, and packet is a TCP or UDP packet
– or, the **load-balance force-l4-hashing** command is configured.

  - Layer-2 packets with an MPLS payload: source MAC address and destination MAC address, VLAN ID, Inner VLAN ID (for double-tagged packets), Ethertype, and up to 3 MPLS Labels.

NOTE
For double-tagged packets, Ethertype is not used and the TPID of the inner TAG must be 0x8100 to be considered a double-tagged packet.

  - GRE encapsulated IPv4 traffic: source MAC address and destination MAC address, IPv4 source and destination address, IPv4 protocol number. VLAN-ID, and TCP or UDP source port and TCP or UDP destination port. Also, inner IPv4 source address and inner destination IPv4 address are used if there is at least one GRE or IPv6 tunnel configured.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:
– Packet is non-fragmented and without option, and packet is a TCP or UDP packet
– or, the **load-balance force-l4-hashing** command is configured.

- IPv6 tunnel encapsulated IPv6 traffic: source MAC address and destination MAC address, IPv6 source and destination address, TCP or UDP source port and TCP or UDP destination port, and IPv6 next header. and VLAN-ID.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:
– Packet is a TCP or UDP packet
– or, the **load-balance force-l4-hashing** command is configured.
For 6over4 encapsulated packets, inner IPv6 destination address and inner IPv6 source address
are used if there is at least one GRE or IPv6 tunnel configured.

- Layer-2, non-IPv4.IPv6 or non-MPLS packets: source MAC address, destination MAC address, VLAN ID, Ethertype and Inner VLAN ID (for double-tagged packets).

- For L3-Routing, the hash index is based on the following:
  - IPv4 or IPv6 traffic: source MAC address and destination MAC address, IPv4v6 source and destination address, VLAN-ID, IPv4 protocol number or IPv6 next header and TCP or UDP source port and TCP or UDP destination port.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:
– Packet is non-fragmented and without option, and packet is a TCP or UDP packet
– or, the **load-balance force-l4-hashing** command is configured.

  - GRE encapsulated IPv4 traffic: source MAC address and destination MAC address, IPv4 source and destination address, IPv4 protocol number. VLAN-ID, and TCP or UDP source port and TCP or UDP destination port. Also, inner IPv4 source address and inner destination IPv4 address are used if there is at least one GRE or IPv6 tunnel configured.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:
– Packet is non-fragmented and without option, and packet is a TCP or UDP packet
– or, the **load-balance force-l4-hashing** command is configured.

  - IPv6 tunnel encapsulated IPv6 traffic: source MAC address and destination MAC address, IPv6 source and destination address, TCP or UDP source port and TCP or UDP destination port, and IPv6 next header. and VLAN-ID.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:
– Packet is a TCP or UDP packet
– or, the **load-balance force-l4-hashing** command is configured.
For 6over4 encapsulated packets, inner IPv6 destination address and inner IPv6 source address
are used if there is at least one GRE or IPv6 tunnel configured.

- For MPLS Switching, the hash index is based on the following:
  - L2VPN traffic: outer source MAC address and outer destination MAC address, up to two MPLS Labels, VLAN ID, inner source MAC address and inner destination MAC address, If packet payload is an IPv4 or v6 packet: IPv4v6 source and destination address, IPv4 Protocol Number or IPv6 Next Header ID of the payload are used.

- L3VPN traffic or IP shortcut traffic: outer source MAC address and outer destination MAC address, VLAN ID, inner source IPv4v6 address and inner destination IPv4v6 address, IPv4 Protocol Number or IPv6 Next Header ID, TCP source port and TCP destination port, UDP source port and UDP destination port, and up to two MPLS Labels.
- MPLS packets with 3 labels: outer source MAC address and outer destination MAC address, VLAN ID, and all 3 MPLS Labels.

NOTE
For transit LSR's please note the following:
The **load-balance speculate-mpls-ip** command must be active. It is on by default.

If the **load-balance speculate-mpls-ip** command has been configured to be inactive, and the **load-balance speculate-mpls-enet** command is active, the packet will be processed like an L2VPN packet.

If both commands are configured to be inactive, no inner Layer-2 or Layer-3 headers are considered
but up to 3 MPLS labels are used for hashing.

Options for hash based load sharing

The following options can be used to refine the hash calculations used for LAGs:

- Speculate UDP or TCP Headers
- Mask Layer-4 Source and Destination Port Information
- Hash Diversification

Each of these options when configured apply to both IP Load Sharing and LAG Load sharing. They are described in detail in "Options for IP load sharing and LAGs" on page 734.

Load sharing for MPLS LAGs

Load sharing on MPLS LAG involves traffic flows that include the MPLS Inner and Outer Labels. These can be used exclusively or in combination with the IP and MAC source and destination addresses to determine the LAG index for a traffic flow. The following additional CLI commands can be added to restrict the hashing to just **MPLS-ip** and **MPLS-enet** respectively.

Using IP source and destination addresses for load sharing

You can use the **load-balance speculate-mpls-ip** command to include the IP source and destination addresses in the calculation of the LAG index for a traffic flow within MPLS LAGs, as shown in the following.

```
NetIron(config)# load-balance speculate-mpls-ip all
```

Syntax: [no] load-balance speculate-mpls-ip [all | <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

The **all** option applies the command to all ports within the router.

Specifying a slot number using the <*slot-number*> variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the <*slot-number*> and <*np-id*> variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE
The **load-balance speculate-mpls-ip** command will hash only on the IP portion.

Using MAC source and destination addresses for load sharing

You can use the **load-balance speculate-mpls-enet** command to include the MAC source and destination addresses in the calculation of the LAG index for a traffic flow within MPLS LAGs, as shown in the following.

```
NetIron(config)# load-balance speculate-mpls-enet all
```

Syntax: [no] load-balance speculate-mpls-enet [all | <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

The **all** option applies the command to all ports within the router.

Specifying a slot number using the <*slot-number*> variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the <*slot-number*> and <*np-id*> variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE
The **load-balance speculate-mpls–enet** command will hash only on the Ethernet header portion.

Per packet server LAG load sharing

Per packet LAG load balancing is a type of LAG that load balances traffic on a per-packet basis, as compared to traditional server LAG load-balancing which balances traffic based on packet content such as source or destination addresses. In per packet server LAG load balancing, the packet processor (PPCR) on each module selects a port in the per packet server LAG to forward traffic in a round-robin fashion. For example, if the first port of the per packet server LAG is currently selected, the second port of the per-packet server LAG will be used next, and so on. Consequently, traffic is evenly distributed among all of the ports that are configured in a per packet server LAG.

Traffic that can be forwarded out of a per-packet LAG includes L2 switching traffic, L3 routing traffic, L3VPN (2547) traffic, VLL and VPLS traffic.

Configuring a LAG

The following configuration procedures are used to configure a LAG. Depending upon whether you are configuring a static, dynamic or keep-alive LAG, the configuration procedures may or may not apply as described:

- **Creating a Link Aggregation Group** – Required for all static, dynamic or keep alive LAGs.
- **Adding Ports to a LAG** – Required for all static, dynamic, or keep alive LAGs. A keep alive LAG contains only one port with static and dynamic LAGs can have 2 to 32 ports.
- **Configuring the Primary Port for a LAG** – Required for all static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.
- **Configuring the Load Sharing Type** – Optional for all static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.

- **Specifying the LAG Threshold for a LAG Group** – Optional for static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.
- **Configuring LACP Port Priority** – Optional for dynamic and keep alive LAGs. Because static LAGs do not support LACP, it is unnecessary to configure this parameter.
- **Configuring an LACP Timeout** – Optional for dynamic and keep alive LAGs. Because static LAGs do not support LACP, it is unnecessary to configure this parameter.

Creating a Link Aggregation Group (LAG) using the LAG ID option

Before setting-up ports or configuring any other aspects of a LAG, you must create it first.

You can either assign a LAG **ID** explicitly or it will be automatically generated by the system. The LAG **ID** stays the same across system reload and hitless upgrade.

The command to configure LAGs allows explicit configuration of the LAG ID for static and dynamic LAGs.

To create a LAG with the LAG ID option, enter a command such as the following.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)#
```

Syntax: [no] lag <*name*> **[static | dynamic] [id** <*number*>**]**

The **ID** parameter is optional. The value of the **ID** parameter that you can enter is from 1 to 256. If you do not enter a LAG **ID**, the system will generate one automatically. Once the LAG **ID** is generated the system will save it in the configuration file along with the LAG name, therefore the value will stay the same across system reload.

NOTE
The LAG **ID** parmeter is for static and dynamic LAGs only. No explicit configuration of a LAG **ID** is allowed on keepalive LAGs.

The **static** parameter specifies that the LAG with the name specified by the <*lag-name*> variable will be configured as a static LAG.

The **dynamic** option specifies that the LAG with the name specified by the <*lag-name*> variable will be configured as a dynamic LAG.

Configuration considerations

LAG **ID**s are unique for each LAG in the system. The same LAG **ID** cannot be assigned to two or more different LAGs. If a LAG **ID** is already used, the CLI will reject the new LAG configuration and display an error message that suggests the next available LAG **ID** that can be used.

Example

```
NetIron(config)#lag lag3 static id 123
Error: LAG id 123 is already used. The next available LAG id is 2.
```

Example : LAG configured with LAG id 124.

```
!
lag "lag1" static id 124
 ports ethernet 1/2 to 1/3
 primary-port 1/3
 deploy
!
```

Example : show lag command and the output.

```
NetIron(config)# show lag
Total number of LAGs:          1
Total number of deployed LAGs: 1
Total number of s created:1 (127 available)
LACP System Priority / ID:     1 / 0000.0001.c000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "lag1" ID 124 (static Deployed) ===
LAG Configuration:
   Ports:         ethe 1/2 to 1/3
   Port Count:    2
   Primary Port:  1/3
    Type:    hash-based
Deployment:   ID 124, Active Primary 1/2
Port  Link L2 State  Dupl Speed  Tag Priori MAC Name
1/2   Up   Forward   Full 10G   124   No  level0 0000.0001.c002
1/3   Up   Forward   Full 10G   124   No  level0 0000.0001.c002
```

Creating a keepalive LAG

To create a **keep-alive** LAG, enter the following.

```
NetIron(config)# lag lag1 keep-alive
```

Syntax: [no] lag <*name*> **keep-alive**

The **keep-alive** option specifies that the LAG with the name specified by the <*lag-name*> variable will be configured a keep-alive LAG. The keep-alive LAG option allows you to configure a LAG for use in keep alive applications similar to the UDLD feature.

Adding Ports to a LAG or Deleting Ports from a LAG

A static or dynamic LAG can consist of from 2 to 32 ports of the same type and speed that are on any interface module within the PowerConnect chassis. A keep alive LAG consists of only one port.

To configure the static LAG named “blue” with two ports, use the following command:

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# ports ethernet 3/1 ethernet 7/2
```

Syntax: [no] ports ethernet <*slot/port*> **| pos** <*slot/port*> **[to** <*slot/port*> **] [ethernet** <*slot/port*> **| pos** <*slot/port*> **]**

The ports added to a LAG can be of type **ethernet** or **pos** as specified for the **slot/port** where they reside. The ports can be added to the LAG sequentially as shown in the following example:

```
NetIron(config-lag-blue)# ports ethernet 3/1 ethernet 7/2 ethernet 4/3 ethernet
3/4
```

A range of ports from a single interface module can be specified. In the following example, Ethernet ports 1, 2, 3 and 4 on the interface module in slot 3 are configured in a single LAG:

```
NetIron(config-lag-blue)# ports ethernet 3/1 to 3/4
```

Additionally, you can mix a range of ports from one interface module with individual ports from other interface modules to form a LAG as shown in the following:

```
NetIron(config-lag-blue)# ports ethernet 3/1 to 3/4 ethernet 10/2
```

Using the **no** option allows you to remove ports from a LAG. For example, you can remove port 3/4 from the LAG created above, as shown in the following:

```
NetIron(config-lag-blue)# no ports ethernet 3/4
```

Ports can be added to an undeployed LAG or to currently deployed LAG using the commands described. For special considerations when adding ports to or deleting ports from a currently deployed LAG, refer to the following sections:

- "Adding a Port to Currently Deployed LAG" on page 224
- "Deleting a Port from a Currently Deployed LAG" on page 224

Configuring the primary port for a LAG

The primary port must be explicitly assigned using the **primary port** command.

To designate the primary port for the static LAG "blue", use the following command.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# primary port 3/2
```

Syntax: [no] primary port <*slot/port*>

Once a primary port has been configured for a LAG, all configurations that apply to the primary port are applied to the other ports in the LAG.

NOTE
This configuration is only applicable for configuration of a static or dynamic LAGs.

Configuring load sharing type

Individual LAGs can be configured to perform load sharing over the ports in the LAG using either a hash based or per packet method, as shown in the following.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# trunk-type hash-based
```

Syntax: [no] trunk-type hash-based | per-packet

NOTE
This configuration is only applicable for configuration of a static or dynamic LAGs.

Specifying the LAG threshold for a LAG group

You can configure the PowerConnect router to disable all of the ports in a LAG group when the number of active member ports drops below a specified threshold value. For example, if a LAG group has 8 ports, and the threshold for the LAG group is 5, then the LAG group is disabled if the number of available ports in the LAG group drops below 5. If the LAG group is disabled, then traffic is forwarded over a different link or LAG group.

NOTE
This configuration is only applicable for configuration of a static or dynamic LAGs.

For example, the following commands establish a LAG group consisting of 4 ports, then establish a threshold for this LAG group of 3 ports.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# ports ethernet 3/1 to 3/4
NetIron(config-lag-blue)# trunk-threshold 3
```

In this example, if the number of active ports drops below 3, then all the ports in the LAG group are disabled.

Syntax: [no] trunk-threshold <*number*>

You can specify a threshold from 1 (the default) up to the number of ports in the LAG group.

When a LAG is shut down because the number of ports drops below the configured threshold, the LAG is kept intact and it is re-enabled if enough ports become active to reach the threshold.

NOTE
The **trunk-threshold** command should be configured only at one end of the trunk. If it is set on both sides, link failures will result in race-conditions and the will not function properly.

```
NetIron(config)# lag blue dynamic
NetIron(config-lag-blue)# lacp-port-priority 100000
```

Syntax: [no] lacp-port-priority <*slot/port*> <*number*>

NOTE
This configuration is only applicable for configuration of a dynamic or keep-alive LAGs.

Configuring an LACP timeout

In a dynamic or keep-alive LAG, a port's timeout can be configured as short (3 seconds) or long (90 seconds). After you configure a port timeout, the port remains in that timeout mode whether it is up or down and whether or not it is part of a LAG.

All the ports in a LAG should have the same timeout mode. This requirement is checked when the LAG is enabled on the ports. For example, to configure a port for a short LACP timeout, use the following command.

```
NetIron(config)# lag blue dynamic
NetIron(config-if-e10000-2/1)# lacp-timeout short
```

Syntax: [no] lacp-timeout [long | short]

To delete the configuration, use the **no** form of this command.

The **long** keyword configures the port for the long timeout mode–90 seconds. With the long timeout, an LACPDU is sent every 30 seconds. If no response comes from its partner after 3 LACPDUs are sent, a timeout event occurs, and the LACP state machine transition to the appropriate state based on its current state.

The **short** keyword configures the port for the short timeout mode—3 seconds. In the short timeout configuration, an LACPDU is sent every second. If no response comes from its partner after 3 LACPDUs are sent, a timeout event occurs, and the LACP state machine transitions to the appropriate state based on its current state.

If you specify neither **long** nor **short**, the state machine operates based on the standard IEEE specification as its default behavior. The original IEEE specification says that the state machine starts with short the timeout and moves to the long timeout after the LAG is established. However, sometimes a vendor's implementation always uses either the short timeout or the long timeout without changing the timeout. Dell provides this command so that you can configure Dell devices to interoperate with other vendor's devices.

NOTE
This configuration is applicable to the configuration of dynamic or keep-alive LAGs only.

Deploying a LAG

After configuring a LAG, you must explicitly enable it before it begins aggregating traffic. This task is accomplished by executing the **deploy** command within the LAG configuration. After the **deploy** command runs, the LAG is in the aggregating mode. Only the primary port within the LAG is available at the individual interface level. Any configuration performed on the primary port applies to all ports within the LAG. The running configuration will no longer display deployed LAG ports other than the primary port.

To deploy a LAG, at least one port must be in the LAG and the primary port must be specified for non keep-alive LAGs. After a non keep-alive LAG is deployed, a LAG is formed. If there is only one port in the LAG, a single port LAG is formed. For a dynamic LAG, LACP is stared for each LAG port. For a keep-alive LAG, no LAG is formed and LACP is started on the LAG port.

You can deploy a LAG as shown in the following for the "blue" LAG.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# deploy
```

Syntax: [no] deploy [forced | passive]

When the **deploy** command is executed:

For a static and dynamic LAGs, the current LAG veto mechanism is invoked to make sure the LAG can be formed. If the LAG is not vetoed, a LAG is formed with all the ports in the LAG.

For dynamic LAGs, LACP is activated on all LAG ports. When activating LACP, use active mode if **passive** is not specified; otherwise, use **passive** mode.

For a keep-alive LAGs, no LAG is formed, and LACP is started on the LAG port.

Once the **deploy** command is issued, all LAG ports will behave like a single port.

If the **no deploy** command is executed, the LAG is removed. For dynamic LAGs, LACP is de-activated on all of the LAG ports.

If the **no deploy** command is issued and more than 1 LAG port is not disabled the command is aborted and the following error message is displayed: “Error 2 or more ports in the LAG are not disabled, un-deploy this LAG may form a loop - aborted.” Using the **forced** keyword with the **no deploy** command in the previous situation, the un-deployment of the LAG is executed.

Commands available under LAG once it is deployed

Once a LAG has been deployed, the following configurations can be performed on the deployed LAG:

- Configuring ACL-based Mirroring
- Disabling Ports within a LAG
- Enabling Ports within a LAG
- Monitoring and Individual LAG Port
- Assigning a name to a port within a LAG
- Enabling sFlow Forwarding on a port within a LAG
- Setting the sFlow Sampling Rate for a port within a LAG

Configuring ACL-based mirroring

To configure ACL-based mirroring for all ports in a LAG, configure it on the primary port of the LAG at the interface configuration level (see “ACL-based inbound mirroring” on page 148). ACL-based mirroring can be configured for an individual member port within a LAG by using the **acl-mirror-port** command, as shown in the following.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# deploy
NetIron(config-lag-blue)# acl-mirror-port ethe-port-monitored 3/1 ethernet 3/2
```

In this example, traffic on Ethernet port 3/1 (a member port of LAG “blue”) will be mirrored to Ethernet port 3/2.

Syntax: [**no**] **acl-mirror-port** { **ethe-port-monitored** <*slot/port*> | **named-port-monitored** <*name*> | **pos-port-monitored** <*slot/port*> } **ethernet** <*slot/port*>

Use the **ethe-port-monitored** option with the appropriate <*slot/port*> variable to specify an Ethernet port for which you want to provide ACL mirroring.

Use the **named-port-monitored** option with the appropriate <*slot/port*> variable to specify a named port for which you want to provide ACL mirroring.

Use the **pos-port-monitored** option with the appropriate <*slot/port*> variable to specify a Packet-over-SONET port for which you want to provide ACL mirroring.

The **ethernet** keyword precedes the <*slot/port*> variable identifying the port which will receive the mirrored packets.

NOTE
A port with ACL-based mirroring already configured on it cannot be added to a LAG, and a LAG cannot be deployed if any of its member ports has ACL-based mirroring. To use ACL-based mirroring on a LAG member port, deploy the LAG, then configure mirroring on the member port. If a port is removed from a LAG, ACL-based mirroring will be removed from that port, and if a LAG is deleted mirroring will be removed from all member ports.

Disabling ports within a LAG

You can disable an individual port within a LAG using the disable command within the LAG configuration as shown in the following.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# deploy
NetIron(config-lag-blue)# disable ethernet 3/1
```

Syntax: [no] disable ethernet [slot/port] | named [name] |pos [slot/port]

Use the **ethernet** option with the appropriate **[slot/port]** variable to specify a Ethernet port within the LAG that you want to disable.

Use the **named** option with the appropriate **[slot/port]** variable to specify a named port within the LAG that you want to disable.

Use the **pos** option with the appropriate **[slot/port]** variable to specify a Packet-over-SONET port within the LAG that you to disable.

Enabling ports within a LAG

You can enable an individual port within a LAG using the enable command within the LAG configuration as shown in the following.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# deploy
NetIron(config-lag-blue)# enable ethernet 3/1
```

Syntax: [no] enable ethernet [slot/port] | named [name] |pos [slot/port]

Use the **ethernet** option with the appropriate **[slot/port]** variable to specify a Ethernet port within the LAG that you want to enable.

Use the **named** option with the appropriate **[slot/port]** variable to specify a named port within the LAG that you want to enable.

Use the **pos** option with the appropriate **[slot/port]** variable to specify a Packet-over-SONET port within the LAG that you to enable.

Adding a Port to Currently Deployed LAG

Ports can be added to a currently deployed LAG. Adding a port to a deployed LAG uses the same procedures as described in "Adding Ports to a LAG or Deleting Ports from a LAG" on page 219. When you add ports to a deployed LAG, the MAC address of the port being added is changed to that of the primary port of the LAG to which it is being added.

Deleting a Port from a Currently Deployed LAG

Ports can be deleted from a currently deployed LAG. Deleting a port in a currently deployed LAG uses the same procedures as described in "Adding Ports to a LAG or Deleting Ports from a LAG" on page 219. However, when deleting ports from a currently deployed LAG you must consider the following:

- The primary port cannot be removed.

- If removal of a port will result in the trunk threshold value becoming greater than the number of ports in the LAG, the port deletion will be rejected.
- The port being deleted must be in the “disabled” state or you must use the forced option (as described in the following command syntax) when deleting it from a currently deployed LAG. Otherwise, the deletion request will be denied and the following error message will be displayed: “Error: ports to be deleted from the deployed LAG are not disabled, deleting these ports from the LAG may form a loop – aborted.”

To delete port 3/1 which is in the “enabled” state from a currently deployed LAG named “blue”, use the following command:

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# no ports ethernet 3/1 forced
```

Syntax: **no ports ethernet** <*slot/port*> **|** **pos** <*slot/port*> **[to** <*slot/port*> **] [ethernet** <*slot/port*> **|** **pos** <*slot/port*> **] [forced]**

This command operates as described in “Adding Ports to a LAG or Deleting Ports from a LAG” on page 219 except for the **forced** option which is described in the following:

The **forced** option to the **no ports** command deletes a port from a currently deployed LAG even if it is currently in the “enabled state”. Because deleting an enabled port from a currently deployed LAG can cause a loop to be formed, we recommend that you disable any port being removed from a LAG before removing it. Only use the **forced** option when you are confident that a loop will not be created in your network topology.

NOTE
When a port is deleted from a currently deployed LAG, the MAC address of the port is changed back to its original value.

Monitoring an individual LAG port

By default, when you monitor the primary port in a LAG group, aggregated traffic for all the ports in the LAG is copied to the mirror port. You can configure the device to monitor individual ports in a LAG including Ethernet, POS, or named ports. You can monitor the primary port or another member port individually.

NOTE
You can use only one mirror port for each monitored LAG port. To monitor traffic on an individual port in a LAG group, enter commands such as the following.

This command enables monitoring of an individual port within a LAG.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# deploy
NetIron(config-lag-blue)# monitor ethe-port-monitored 3/1 ethernet 10/3 both
```

Syntax: **[no] monitor ethe-port-monitored [slot/port] | named-port-monitored [name] |pos-port-monitored [slot/port] ethernet [slot/port] [input | output | both]**

Use the **ethe-port-monitored** option with the appropriate **[slot/port]** variable to specify a Ethernet port within the LAG that you want to monitor.

Use the **named-port-monitored** option with the appropriate **[slot/port]** variable to specify a named port within the LAG that you want monitor.

Use the **pos-port-monitored** option with the appropriate **[slot/port]** variable to specify a Packet-over-SONET port within the LAG that you want to monitor.

The **ethernet** <*slot/port*> parameter specifies the port to which the traffic analyzer is attached. While a POS port can be monitored, a POS port cannot be configured as a mirror port. Therefore, traffic monitored from a POS port must be mirrored through an Ethernet port.

The **input | output | both** parameters specify the traffic direction to be monitored.

Assigning a name to a port within a LAG

You can assign a name to an individual port within a LAG using the **port-name** command within the LAG configuration as shown in the following.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# deploy
NetIron(config-lag-blue)# port-name orange ethernet 3/1
```

Syntax: [no] port-name <*text*> **ethernet [slot/port] | pos [slot/port]**

The <*text*> variable specifies the port name. The name can be up to 50 characters long.

Use the **ethernet** option with the appropriate **[slot/port]** variable to apply the specified name to an Ethernet port within the LAG.

Use the **pos** option with the appropriate **[slot/port]** variable to apply the specified name to a Packet-over-SONET port within the LAG.

Refer to "Allowable characters for LAG names" on page 15 for additional information on on LAG naming conventions.

Enabling sFlow forwarding on a port in a LAG

You can enable sFlow forwarding on an individual port within a LAG using the **sflow-forwarding** command within the LAG configuration as shown in the following.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# deploy
NetIron(config-lag-blue)# sflow-forwarding ethernet 3/1
```

Syntax: [no] sflow-forwarding ethernet [slot/port] | port-name [text] |pos [slot/port]

Use the **ethernet** option with the appropriate **[slot/port]** variable to specify a Ethernet port within the LAG that you want to enable sFlow forwarding for.

Use the **port-name** option with the appropriate **[text]** variable to specify a named port within the LAG that you want to enable sFlow forwarding for.

Use the **pos** option with the appropriate **[slot/port]** variable to specify a Packet-over-SONET port within the LAG that you want to enable sFlow forwarding for.

Setting the sFlow sampling rate for a port in a LAG

You can set the sFlow sampling rate for an individual port within a LAG using the **sflow-subsampling** command within the LAG configuration as shown in the following.

```
NetIron(config)# lag blue static
NetIron(config-lag-blue)# deploy
NetIron(config-lag-blue)# sflow-subsampling ethernet 3/1 512
```

Syntax: [no] sflow-subsampling ethernet [slot/port] | port-name [text] |pos [slot/port] <*num*>

Use the **ethernet** option with the appropriate **[slot/port]** variable to specify the Ethernet port within the LAG that you want to configure the sampling rate for.

Use the **port-name** option with the appropriate **[text]** variable to specify the named port within the LAG that you want to configure the sampling rate for.

Use the **pos** option with the appropriate **[slot/port]** variable to specify the Packet-over-SONET port within the LAG that you want to configure the sampling rate for.

The <*num*> variable specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. This can be a value between 512 - 1048576.

Configuring a dynamic LAG within a VRF

In release 3.8.00, support was added for dynamic LAGs within a VRF. When configuring a dynamic LAG within a VRF, the following must be considered:

- The dynamic LAG must be configured before adding it to a VRF.
- Before the LAG is deployed, all members must be in the default VRF.
- After the LAG is deployed, all LAG ports are in the LACP BLOCK state until the LACP protocol completes negotiation with the other end of the LAG.
- Once the LACP protocol negotiation is completed with the other end of the LAG, all the LAG ports are set to the FORWARD state.
- When a dynamic LAG within a VRF is undeployed, the primary port will stay in the VRF where the LAG was configured and the secondary ports of the LAG will return to the default VRF.

The following example uses the LAG and VRF commands to configure a LAG within a VRF.

```
NetIron(config)# lag red dynamic
NetIron(config-lag-red)# primary port 3/2
NetIron(config-lag-red)# ports ethernet 3/1 ethernet 7/2
NetIron(config-lag-red)# exit
NetIron(config)# interface ethernet 3/2
NetIron(config-if-e10000-3/2)# ip vrf forwarding VPN1
NetIron(config)# lag red dynamic
NetIron(config-lag-red)# deploy
```

Displaying LAG information

You can display LAG information for a PowerConnect router in either a **full** or **brief** mode.

The following example displays the **brief** option of the **show lag** command.

```
NetIron# show lag brief
Total number of LAGs:          4
Total number of deployed LAGs: 3
Total number of s created:3 (125 available)
LACP System Priority / ID:     0001 / 0004.80a0.4000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3

LAG           Type    Deploy  Primary  Port List
d1            dynamic  Y    3     32/2     ethe 13/2 to 13/3 ethe 32/2
e             dynamic  Y    1     2/3      ethe 2/1 ethe 2/3 ethe 2/5
p             static   Y    2     3/1      pos 1/2 pos 3/1
s1            static   N    none  32/3     ethe 13/4 ethe 32/3 to 32/4
```

Syntax: show lag brief

Table 40 describes the information displayed by the **show lag brief** command.

The following example displays the full option of the **show lag** command.

```
NetIron# show lag
Total number of LAGs:          4
Total number of deployed LAGs: 3
Total number of s created:3 (125 available)
LACP System Priority / ID:     0001 / 0004.80a0.4000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3

=== LAG "d1" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 13/2 to 13/3 ethe 32/2
   Primary Port:  32/2
    Type:    hash-based
   LACP Key:      104

Deployment:   ID 3, Active Primary 3/2

Port  Link L2 State  Dupl Speed  Tag Priori MAC            Name
3/2   Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
13/3  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
32/2  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9

Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
13/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
13/3       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
32/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope

=== LAG "e" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 2/1 ethe 2/3 ethe 2/5
   Primary Port:  2/3
    Type:    hash-based
   LACP Key:      105

Deployment:   ID 1

Port  Link L2 State  Dupl Speed  Tag Priori MAC            Name
2/1   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/3   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/5   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a

Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
2/1        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/3        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/5        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
```

Syntax: show lag <*lag-name*> **[deployed] [dynamic] [keep-alive] [static]**

Using command this without options displays information for all LAGs configured on the router.

The <*lag-name*> variable allows you to limit the display to information for a specific LAG.

The **deployed** option limits the display to LAGs that are currently deployed.

The **dynamic** option limits the display to dynamic LAGs.

The **keep-alive** option limits the display to keep alive LAGs.

The **deployed** option limits the display to static LAGs.

Table 40 describes the information displayed by the **show lag** command.

TABLE 40 Show LAG information

This field...	Displays...
Total number of LAGS	The total number of LAGs that have been configured on the router.
Total number of Deployed LAGS	The total number of LAGs on the router that are currently deployed.
Total number of LAGs Created	The total number of LAGs that have been created on the LAG. The total number of LAGs available are shown also. Since keep-alive LAGs do not use a LAG ID, they are not listed and do not subtract for the number of LAGs available.
LACP System Priority /ID	The system priority configured for the router. The ID is the system priority which is the base MAC address of the router.
LACP Long timeout	The number of seconds used for the LACP Long timeout mode. This is only applicable for dynamic or keep-alive LAGs.
LACP Short timeout	The number of seconds used for the LACP Short timeout mode. This is only applicable for dynamic or keep-alive LAGs.
The following information is displayed per-LAG in the show lag brief command.	
LAG	The name of the LAG.
Type	The configured type of the LAG: static, dynamic, or keep-alive
Deploy	Status of LAG deployment: Y – yes, LAG is deployed. N – no, LAG is not deployed.
LAG	The LAG ID number.
Primary	The primary port of the LAG.
Port List	The list of ports that are configured in the LAG.
The following information is displayed per-LAG the show lag command for each LAG configured.	
LAG Configuration	
Ports:	List of ports configured with the LAG.
Primary Port:	The primary port configured on the LAG.
LAG Type:	The load sharing method configured for the LAG: either hash-based or per-packet.
LACP Key	The link aggregation key for the LAG.
Deployment	
LAG ID	The LAG ID number.
Active Primary	The port within the LAG where most protocol packets are transmitted. This is not the same as the configured Primary Port of the LAG.
Port	The chassis slot and port number of the interface.
Link	The status of the link which can be one of the following: • up • down
L2 State	The L2 state for the port.
Dupl	The duplex state of the port, which can be one of the following: • Full • Half • None

TABLE 40 Show LAG information (Continued)

This field...	Displays...
Speed	The bandwidth of the interface.
LAG	The LAG ID of the port.
Tag	Indicates whether the ports have 802.1q VLAN tagging. The value can be Yes or No.
Priori	Indicates the Quality of Service (QoS) priority of the ports. The priority can be a value from 0 – 7.
MAC	The MAC address of the port.
Name	The name (if any) configured for the port.
Sys P	Lists the system priority configured for the device.
Port P	Lists the port's link aggregation priority.
Key	Lists the link aggregation key.
Act	Indicates the link aggregation mode, which can be one of the following: • No – The mode is passive on the port. If link aggregation is enabled (and the mode is passive), the port can send and receive LACPDU messages to participate in negotiation of an aggregate link initiated by another port, but cannot search for a link aggregation port or initiate negotiation of an aggregate link. • Yes – The mode is active. The port can send and receive LACPDU messages.
Tio	Indicates the timeout value of the port. The timeout value can be one of the following: • L – Long. The LAG group has already been formed and the port is therefore using a longer message timeout for the LACPDU messages exchanged with the remote port. Typically, these messages are used as confirmation of the health of the aggregate link. • S – Short. The port has just started the LACPDU message exchange process with the port at the other end of the link. The S timeout value also can mean that the link aggregation information received from the remote port has expired and the ports are starting a new information exchange.
Agg	Indicates the link aggregation state of the port. The state can be one of the following: • Agg – Link aggregation is enabled on the port. • No – Link aggregation is disabled on the port.
Syn	Indicates the synchronization state of the port. The state can be one of the following: • No – The port is out of sync with the remote port. The port does not understand the status of the LACPDU process and is not prepared to enter a LAG link. • Syn – The port is in sync with the remote port. The port understands the status of the LACPDU message exchange process, and therefore knows the LAG group to which it belongs, the link aggregation state of the remote port, and so on.
Col	Indicates the collection state of the port, which determines whether the port is ready to send traffic over the LAG link: • Col – The port is ready to send traffic over the LAG link. • No – The port is not ready to send traffic over the LAG link.
Dis	Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the LAG link. • Dis – The port is ready to receive traffic over the LAG link. • No – The port is not ready to receive traffic over the LAG link.

TABLE 40 Show LAG information (Continued)

This field...	Displays...
Def	Indicates whether the port is using default link aggregation values. The port uses default values if it has not received link aggregation information through LACP from the port at the remote end of the link. This field can have one of the following values: • Def – The port has not received link aggregation values from the port at the other end of the link and is therefore using its default link aggregation LACP settings. • No – The port has received link aggregation information from the port at the other end of the link and is using the settings negotiated with that port.
Exp	Indicates whether the negotiated link aggregation settings have expired. The settings expire if the port does not receive an LACPDU message from the port at the other end of the link before the message timer expires. This field can have one of the following values: • Exp – The link aggregation settings this port negotiated with the port at the other end of the link have expired. The port is now using its default link aggregation settings. • No – The link aggregation values that this port negotiated with the port at the other end of the link have not expired, so the port is still using the negotiated settings.
Ope	• Ope (operational) - The port is operating normally. • Blo (blocked) - The port is blocked because the adjacent port is not configured with link aggregation or because it is not able to join a LAG group. An LACP port is blocked until it becomes part of a LAG. Also, an LACP is blocked if its state becomes “default”. To unblock the port and bring it to an operational state, enable link aggregation on the adjacent port and ensure that the ports have the same key.

Displaying LAG statistics

You can display LAG statistics for a PowerConnect router in either a **full** or **brief** mode. Full mode is the default and is displayed when the **show statistics lag** command is executed without the **brief** option. The examples below show both options of the **show statistics lag** command.

```
NetIron# show statistics brief lag
LAG                  Packets            Collisions                  Errors
                      [Receive            Transmit]   [Recv  Txmit]      [InErr
OutErr]
        LAG d1          1173              1018        0       0        0       0
         LAG e          1268              1277        0       0        0       0

NetIron# show statistics lag
LAG d1 Counters:
         InOctets             127986          OutOctets             107753
           InPkts               1149            OutPkts                996
  InBroadcastPkts                 0    OutBroadcastPkts                  0
  InMulticastPkts               852    OutMulticastPkts                684
    InUnicastPkts               297      OutUnicastPkts                312
       InDiscards                 0         OutDiscards                  0
         InErrors                 0           OutErrors                  0
     InCollisions                 0       OutCollisions                  0
                                        OutLateCollisions                 0
        Alignment                 0                 FCS                   0
```

```
      GiantPkts                0          ShortPkts                0
    InBitsPerSec               0       OutBitsPerSec               0
    InPktsPerSec               0       OutPktsPerSec               0
   InUtilization            0.0%      OutUtilization            0.0%
```

Syntax: show statistics [brief] lag [<*lag-name*> **]**

VLANs 8

Overview

The following VLAN features are supported by NetIron MLX Series devices.

- VLANs
- VLAN Tagging
- Port-Based VLANs
- Protocol-Based VLANs
- Virtual Routing Interfaces
- Integrated Switch Routing (ISR)
- VLAN Groups
- Super Aggregated VLANs
- 802.1q-in-q Tagging
- 802.1q Tag-type Translation
- Assigning a VLAN Priority
- Allocating Memory for More VLANs or Virtual Routing Interfaces
- Uplink Ports Within a Port-Based VLAN
- Hardware Flooding for Layer 2 Multicast and Broadcast Packets
- Unknown Unicast Flooding on VLAN Ports
- VLAN CPU Protection
- VLAN Byte Counters
- Extended VLAN counters
- multi-port static MAC address
- Transparent VLAN Flooding
- VLAN ID Range - 1-4090

A Virtual Local Area Network (VLAN) lets you segment traffic in a network by placing ports and interfaces into separate broadcast domains. Each broadcast domain is uniquely identified by a VLAN ID. These broadcast domains can span multiple devices.

The PowerConnect router supports two types of VLANs: *port-based VLANs* and *protocol-based VLANs*. A port-based VLAN consists of interfaces that constitute a Layer 2 broadcast domain. (Protocol-based VLANs are described in "Protocol-based VLANs" on page 237.) By default, all interfaces on a PowerConnect router are members of the *default* VLAN, which is VLAN 1. Thus, by default, all interfaces on all devices on a network constitute a single Layer 2 broadcast domain. Once you create a port-based VLAN and assign an interface to that VLAN, that interface is automatically removed from the default VLAN if the interface is assigned to the VLAN as an untagged interface. If the interface is assigned as a tagged interface, then the interface is a member of both the default VLAN, and the VLAN it is assigned to.

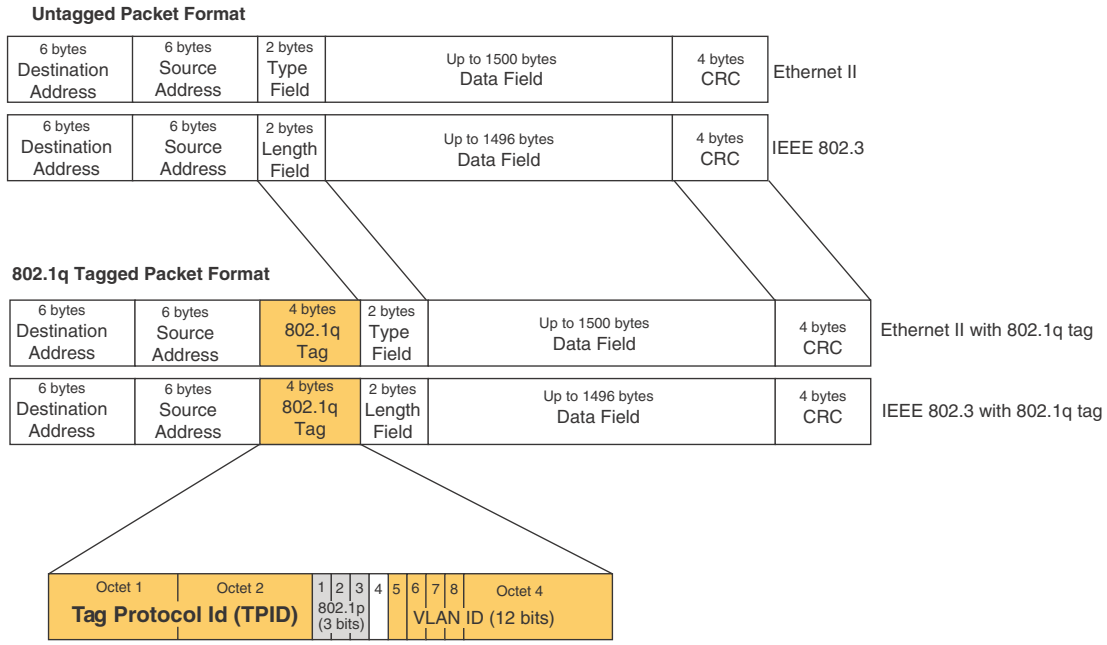
Tagged, untagged, and dual mode ports

Interfaces assigned to port-based VLANs can be defined as untagged, tagged, and dual-mode ports. An untagged port is a member of only one VLAN, while a tagged port can be a member of more than one VLAN. Thus a tagged port can be a member of more than one broadcast domain. Dual-mode ports are configured by adding one or more tagged VLANs and one untagged VLAN to a port.

Tagged ports allow the PowerConnect router to add a four-byte 802.1q tag to the packet. 802.1q tagging is an IEEE standard that allows a networking device to add information to Layer 2 packets. This information identifies the VLAN membership of the packet, as well as the VLAN ID of the VLAN from which the packet is sent. Furthermore, the default tag value of the 802.1q tag is 8100 (hexadecimal). This value comes from the IEEE 802.1q specification. You can change this tag value on a per-port or on a global basis on PowerConnect router if needed to be compatible with other vendors' equipment.

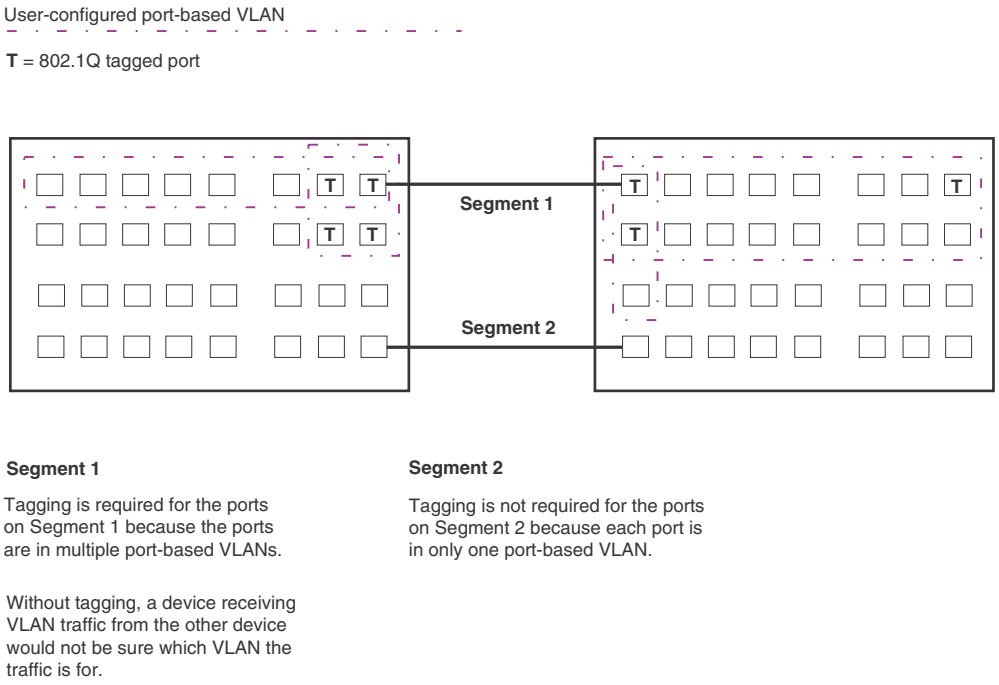
Figure 6 shows the format of packets with and without the 802.1q tag.

FIGURE 6 Packet containing Dell's 802.1QVLAN tag



If you configure a VLAN that spans multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If a port connecting one device to the other is a member of only a single port-based VLAN, tagging is not required. Figure 7 shows an example of two devices that have the same Layer 2 port-based VLANs configured across them. Notice that only one of the VLANs requires tagging.

FIGURE 7 VLANs configured across multiple devices



Protocol-based VLANs

Interfaces that belong to a port-based VLAN can further be divided into Layer 3 broadcast domains by using protocol-based VLANs. Protocol-based VLANs accept broadcasts of a specified protocol type. For example, an IP subnet VLAN accepts broadcasts for the specified IP subnets only. This feature enables you to limit the amount of broadcast traffic to end-stations, servers, and routers.

In a PowerConnect router, you can configure the following protocol-based VLANs within a port-based VLAN:

- **AppleTalk** - The device sends AppleTalk broadcasts to all ports within the AppleTalk protocol VLAN.
- **IP** - The device sends IP broadcasts to all ports within the IP protocol VLAN.
- **IPX** - The device sends IPX broadcasts to all ports within the IPX protocol VLAN.
- **IPv6** - The device sends IPv6 broadcasts to all ports within the IPv6 protocol VLAN.

NOTE
You can configure a protocol-based VLAN as a broadcast domain for IPv6 traffic. When the PowerConnect router receives an IPv6 multicast packet (a packet with 06 in the version field and 0xFF as the beginning of the destination address), the PowerConnect router forwards the packet to all other ports in the VLAN except to the port that received the packet.

Protocol-based VLANs can be configured to have *static* or *excluded* port memberships. Static ports are permanent members of a protocol-based VLAN. They remain active members of the protocol-based VLAN regardless of whether they receive traffic for the VLAN's protocol.

NOTE
The dynamic port membership is not supported on PowerConnect devices.

If you want to exclude certain ports in a port-based VLAN from protocol-based VLANs, the protocol-based VLAN can be explicitly configured to exclude those ports.

VLAN configuration rules

To create any type of VLAN on a PowerConnect router, Layer 2 forwarding must be enabled. When Layer 2 forwarding is enabled, the PowerConnect becomes a switch on all ports for all non-routable protocols.

In addition to this rule, the sections below summarize the rules for configuring VLANs.

NOTE
To enable Layer 2 forwarding, do "**no route-only**".

VLAN ID range

The upper range of VLAN IDs available for user VLANs (including the default VLAN) has been reduced to 4090 (formerly 4094). The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

Tagged VLANs

When configuring VLANs across multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If you are configuring tagged VLANs across multiple devices, make sure all the devices support the same tag format.

VLAN hierarchy

A hierarchy of VLANs exists between the Layer 2 and Layer 3 protocol-based VLANs:

- Port-based VLANs are at the lowest level of the hierarchy.
- Layer 3 protocol-based VLANs are at the highest level of the hierarchy.

As a PowerConnect router receives packets, the VLAN classification starts from the highest level VLAN first. Therefore, if an interface is configured as a member of a port-based VLAN and a protocol-based VLAN, packets coming into the interface are classified as members of the protocol-based VLAN because that VLAN is higher in the VLAN hierarchy.

When a port in a VLAN receives a packet, the device forwards the packet based on the following VLAN hierarchy:

- If it is a Layer 3 packet and the port is a member of a Layer 3 protocol-based VLAN for the packet's protocol, the device forwards the packet on all the Layer 3 protocol-based VLAN ports that have been configured or drops the packet if the port is explicitly excluded from the protocol VLAN.
- If the packet cannot be forwarded based on its VLAN membership types but the packet can be forwarded at Layer 2, the device forwards the packet on all the ports within the receiving port's port-based VLAN.

Multiple VLAN membership rules

The multile VLAN membership rules are listed below:

- A port can belong to multiple, overlapping Layer 2 port-based VLANs only if the port is a tagged port. Packets sent out of a tagged port use an 802.1q-tagged frame.
- A port can belong to multiple, unique, overlapping Layer 3 protocol-based VLANs.
- When both port and protocol-based VLANs are configured on a given device, all protocol-based VLANs must be strictly contained within a port-based VLAN. A protocol-based VLAN cannot include ports from multiple port-based VLANs. This rule is required to ensure that port-based VLANs remain loop-free Layer 2 broadcast domains.
- One of each type of protocol-based VLAN can be configured within each port-based VLAN on the PowerConnect.
- Removing a configured port-based VLAN from a PowerConnect router automatically removes any protocol-based VLAN, or any virtual routing interfaces defined within the port-based VLAN.

Dual-mode default VLAN

As previously described, ports can be defined as dual-mode, which means that they can exist in both tagged and untagged VLANs. As such, they can coexist untagged in the default or a non-default VLAN and be added as a tagged port into non-default VLAN. One way that ports become dual-mode is by adding a port to a non-default, tagged VLAN. The normal behavior is for the port to remain in the default VLAN as an untagged port.

Changing the dual-mode default VLAN behavior

The **no dual-mode-default-vlan** command has been added to change this behavior. This is useful in situations where there is a danger of loops being created if Spanning Tree is not or can not be configured on the default VLAN such as when ports are facing a service provider network and STP BPDUs are not welcome on those ports.

Once the **no dual-mode-default-vlan** command is applied at the global level, a port will not be entered into the dual-mode state by default. If the **no dual-mode-default-vlan** command is configured, when a port is added as tagged to a non-default user-defined VLAN, it is automatically removed from the default VLAN and added to the non-default VLAN as a pure tagged port. Once in this state, a port can only be placed in dual-mode by explicitly configuring it as an untagged port into a non-default VLAN.

When the **no untagged** <*vlan-name*> **command** is applied in dual mode, the port will go into pure tagged mode in contrast to the default operating conditions where the port is automatically placed in the dual mode state in regards to the default VLAN. To change the default condition a a PowerConnect router regarding the dual-mode, default VLAN behavior, enter the **no dual-mode-default-vlan** command as shown in the following.

```
NetIron(config)# no dual-mode-default-vlan
```

Syntax: [no] dual-mode-default-vlan

The default state is for ports added as tagged to a non-default VLAN to remain as untagged ports in the default VLAN and become dual-mode ports.

Using this command with the **no** option, changes the default state and automatically removes a port from the default VLAN when it is added as a tagged port to a non-default VLAN. Using the command without the **no** option, will return the systems behavior to its normal operating condition.

NOTE
When a PowerConnect router is operating in the default state regarding the **no dual-default-vlan** command (which is not configured), syslog messages are generated whenever a port is moved out of the default VLAN. This is normal and expected behavior. Because when the **no dual-default-vlan** command is configured, it is normal operating behavior for a port to be moved out of the default VLAN whenever it enters the dual-mode state syslog messages may be generated when the ports are moved, which is not expected. These messages may be generated in the following situations:

- When a port is added “tagged” into the default VLAN, it is automatically deleted from the default VLAN and a syslog message is generated.
- When a port is in dual-mode, and a user issues the **no untagged** <*port-no*> command within the port VLAN configuration, the port is added back to the default VLAN. However, because the **no dual-default-vlan** command is configured, the port is transitioned out of the default VLAN which generates an additional syslog message.

Restrictions for use of this command

The **no dual-mode-default-vlan** command can only be applied if the router does not currently have any ports configured in dual-mode. If any port is currently in the dual-mode state when the **no dual-mode-default-vlan** command is executed, the command is rejected without it being applied to any ports on the router. Consequently, using the **no dual-mode-default-vlan** command does not cause any action but enables a new behavior for ports that are added to a VLAN.

If there are ports configured into the dual-mode (default VLAN) state, they can be moved from that state by removing untagged ports the default VLAN that also exist as tagged in a non-default VLAN or removing tagged ports from non-default VLANs.

Layer 2 control protocols on VLANs

Layer 2 protocols such as STP, RSTP, Foundry MRP, and VSRP can be enabled on a port-based VLAN.

The Layer 2 state associated with a VLAN and port is determined by the Layer 2 control protocol. Layer 2 broadcasts associated with the VLAN will not be forwarded on this port if the Layer 2 state is not FORWARDING.

It is possible that the control protocol, for example STP, will block one or more ports in a protocol-based VLAN that uses a virtual routing interface to route to other VLANs. For IP protocol and IP subnet VLANs, even though some of the physical ports of the virtual routing interface are blocked, the virtual routing interface can still route as long as at least one port in the virtual routing interface’s protocol-based VLAN is not blocked by STP.

You can also enable Single STP (SSTP) on the device; however, the ports in all VLANs on which SSTP is enabled become members of a single spanning tree. The ports in VLANs on which SSTP is disabled are excluded from the single spanning tree. A VLAN can also be selectively added or removed from the single spanning tree domain.

Virtual interfaces and CPU protection co-existence on VLANs

CPU protection can be configured on VLANs regardless of whether there are virtual-interfaces configured on them (Previously, CPU protection was only configurable if a virtual-interface was not configured on the VLAN).

There is a difference in the behavior of CPU protection in each of the following situations:

- When virtual-interfaces are configured on a VLAN, the CPU-protection is done only on unknown-unicast packets from the VLAN. Multicast and broadcast packets from the VLAN will be sent to the CPU. This allows the CPU to process packets such as ARP and OSPF "hello" packets that may be relevant to the router.
- When virtual-interface is not configured on the VLAN, the CPU-protection is performed for all packets (unknown-unicast, multicast and broadcast) from the CPU.

Configuring port-based VLANs

As explained above, you can place ports into VLANs to segment traffic into broadcast domains. When you create a VLAN, you specify if ports added to that VLAN are tagged or untagged.

NOTE
When adding a port to a VLAN you might get an error message concerning iP routing or IPv6 routing information on the port. If you receive this message, check to see if the port was previously configured for routing protocols such as OSPFv2 or OSPFv3 where the routing protocol was removed globally without first being de-configured on that port. If this is the case, re-enable the routing protocol globally to view the interface configuration and then disable the routing protocol from the port. You can then add the port to the VLAN.

To create a VLAN, perform the tasks listed below.

1. At the global CONFIG level assign an ID to the VLAN.

   ```
   NetIron(config)# vlan 2
   ```

   VLAN IDs can be in the range of 1 – 4090. Use the **no** form of the command to delete the VLAN from the configuration.

   The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

   In addition to a VLAN number, you can assign a name to a VLAN by entering name <*vlan-name*>. Enter up to 32 characters for name.

2. Once a VLAN ID is assigned, the CLI directs you to the VLAN configuration level. At this level, you add ports to that VLAN and specify if the ports are tagged or untagged.

   ```
   NetIron(config-vlan-2)# untag e 1/9 to 1/16
   NetIron(config-vlan-2)# tagged e 1/1 to 1/8
   ```

The example above configures a port-based VLAN, VLAN 2. It adds Ethernet ports 1/9 through 1/16 as untagged ports and ports 1/1 through 1/8 as tagged ports. Since ports 1/9 through 1/16 are untagged, they can be members of VLAN 2 only, while ports 1/1 through 1/8 are tagged ports and can be members of other VLANs.

NOTE
In the configuration above, ports 1/9 – 1/16 are automatically removed from the default VLAN since they are configured as untagged ports; while port 1/1 – 1/8 are still members of the default VLAN.

Syntax: [no] untagged | tagged ethernet <*slot-number*>/<*port-number*> **[to** <*slot-number*>/<*port-number*> **| ethernet** <*slot-number*>/<*port-number*>**]**

The **untagged** and **tagged** parameter removes ports from the default VLAN and puts them in the port-based VLAN. The **untag** command also allows the ports to process packets that do not contain 802.1q tagging.

The **tagged** parameter allows the PowerConnect router to add a four-byte tag 802.1q tag to the packets that go through the tagged ports. It also allows the ports to be members of other VLANs.

Enter the port that you want to assign to the VLAN for the **ethernet** <*slot-number*>/<*port-number*> parameter. You can add LAG group ports to the VLAN by entering the LAG group's primary port. A LAG group's primary port is the port with the lowest number in the LAG group. When you add the LAG group's primary port, all the ports on the LAG group become members of the VLAN.

Use the **no** form of the command to remove the ports from a VLAN.

Example

```
NetIron(config)# vlan 4
NetIron(config-vlan-4)# no untag ethernet 1/11
```

Strictly or explicitly tagging a port

If you want a port to be strictly or explicitly tagged, that port has to be removed from the default VLAN. Enter a command such as the following.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# tagged e 1/1 to 1/8
NetIron(config-vlan-2)# vlan 1
NetIron(config-vlan-1)# no untagged e 1/1 to 1/8
```

Assigning or changing a VLAN priority

You can prioritize traffic on a VLAN by assigning a priority to a VLAN. All packets associated with the VLAN will be classified to the configured priority.

```
NetIron(config-vlan-2)# priority 2
```

Syntax: [no] priority <*num*>

Possible Values: 0 - 7, "0" assigns the lowest priority and "7", the highest priority. The default is "0".

Assigning a different ID to the default VLAN

As stated above, by default, all ports on a PowerConnect router belong to the default VLAN, which is VLAN 1, until it is assigned to a port-based VLAN. The default VLAN port membership is always untagged; however, if you want to use VLAN ID 1 as a configurable VLANs with tagged port members, you can assign a different VLAN ID as the default VLAN. Enter commands such as the following command.

```
NetIron(config)# default-vlan-id 4000
```

Syntax: [no] default-vlan-id <*vlan-id*>

You must specify a VLAN ID that is not already in use. For example, if VLAN 10 exists, do not use "10" as the new VLAN ID for the default VLAN. VLAN IDs are from 1 – 4090. The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

Configuring protocol-based VLANs

Once port-based VLANs are created, you can further segment the broadcast domains by creating protocol-based VLANs, based on Layer 3 protocols. Use the general procedure below for creating protocol-based VLANs.

1. Create the port-based VLAN that contains the interface that you want to segment using Layer 3 protocols.

   ```
   NetIron(config)# vlan 2
   NetIron(config-vlan-2)# untag e 1/9 to 1/16
   NetIron(config-vlan-2)# tagged e 1/1 to 1/8
   ```

2. Under the VLAN configuration level, define the Layer 3 protocol you want to use to segment packets that go through the ports assigned to the port-based VLAN.

   ```
   NetIron(config-vlan-2)# ipv6-proto name Blue
   ```

   Syntax: [no] ip-proto | ipv6-proto | ipx-proto | atalk-proto | other-proto name <*protocol-vlan-name*>

   Enter:

   - **ip-proto** to create a IP protocol VLAN.
   - **ipv6-proto** to create a IPv6 protocol VLAN.
   - **ipx-proto** to create a IPX protocol VLAN.
   - **atalk-proto** to create an Appletalk protocol VLAN.
   - **other-proto** to create a protocol VLAN for protocols other than an IP protocol, IPv6, IPX, or Appletalk protocol.

   Enter **name** <*vlan-name*> if you want to assign a name to the protocol-based VLAN. Enter up to 32 characters for name.

   Use the **no** form of the command to remove the protocol-based VLAN.

3. Assign or exclude specific ports to the protocol-based VLAN.

   ```
   NetIron(config-vlan-group-ipv6-proto)# static e 1/1 e 1/24
   NetIron(config-vlan-group-ipv6-proto)# exclude e 1/2 to 1/4
   ```

   Syntax: [no] static | exclude ethernet <*slot-number*>/<*port-number*> **[to** <*slot-number*>/<*port-number*>]

The **static** ethernet <*slot-number*>/<*port-number*> [to <*slot-number*>/<*port-number*>] parameter adds the specified ports within the port-based VLAN as static ports to the protocol-based VLAN. Packets of the specified protocol will be forwarded on these ports.

The **exclude** ethernet <*slot-number*>/<*port-number*> [to <*slot-number*>/<*port-number*>] parameter excludes the specified ports from the protocol-based VLAN. Packets of the specified protocol will be dropped if received on these ports.

Configuring virtual routing interfaces

The PowerConnect router sends Layer 3 traffic at Layer 2 within a protocol-based VLAN. However, Layer 3 traffic from one protocol-based VLAN to another must be routed. If you want the device to be able to send Layer 3 traffic from one protocol-based VLAN to another on the same router, you must configure a virtual routing interface on each protocol-based VLAN, then configure routing parameters on the virtual routing interfaces.

A *virtual routing interface* is a logical routing interface that the PowerConnect router uses to route Layer 3 protocol traffic between protocol-based VLANs. It is a logical port on which you can configure Layer 3 routing parameters.

For example, to enable a PowerConnect router to route IP traffic from one IP protocol VLAN to another, you must configure a virtual routing interface on each IP protocol VLAN, then configure the appropriate IP routing parameters on each of the virtual routing interfaces.

Example

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# tagged e 1/1 to 1/2
NetIron(config-vlan-2)# router-interface ve 1
```

The PowerConnect router can locally route IP packets between VLANs that are defined within a single router.

If you do not need to further partition the port-based VLAN into protocol-based VLANs, you can define a single virtual routing interface at the port-based VLAN level and enable routing on a single virtual routing interface.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# tagged e 1/1 to 1/2
NetIron(config-vlan-2)# router-interface ve 2
NetIron(config-vlan-2)# exit
NetIron(config)# interface ve 2
NetIron(config-ve-2)# ip address 10.1.1.1/24
```

Syntax: **router-interface ve** <*ve-number*>

Enter 1 to the maximum number of virtual routing interfaces supported on the device for <*ve-number*>.

Integrated Switch Routing (ISR)

Integrated Switch Routing (ISR) feature enables VLANs configured on the PowerConnect router to route Layer 3 traffic from one protocol-based VLAN to another instead of forwarding the traffic to an external router. The VLANs provide Layer 3 broadcast domains for the protocols, but do not in themselves provide routing services. This is true even if the source and destination protocols are on the same device.

ISR eliminates the need for an external router by allowing you to route between VLANs using virtual routing interfaces (ves). You configure a separate virtual routing interface on each VLAN that you want to use to route packets. For example, if you configure two IP protocol VLANs on a PowerConnect router, you can configure a virtual routing interface on each of the IP protocol VLAN, then configure IP routing parameters for the IP protocol VLAN. Thus, the PowerConnect router forwards IP broadcasts within each VLAN at Layer 2 but routes Layer 3 traffic between the VLANs using the virtual routing interfaces.

NOTE
The PowerConnect router uses the lowest MAC address on the device (the MAC address of port 1/1) as the MAC address for all ports within all virtual routing interfaces you configure on the device.

The routing parameters and the syntax for configuring them are the same as when you configure a physical interface for routing (for example, **interface ve 10**). The logical interface allows the PowerConnect router to internally route traffic between the protocol-based VLANs without using physical interfaces.

All the ports within a protocol-based VLAN must be in the same port-based VLAN. The protocol-based VLAN cannot have ports in multiple port-based VLANs, unless the ports in the port-based VLAN to which you add the protocol-based VLAN are 802.1q tagged.

You can configure multiple protocol-based VLANs within the same port-based VLAN. In addition, a port within a port-based VLAN can belong to multiple protocol-based VLANs of the same type or different types. For example, if you have a port-based VLAN that contains ports 1/1 – 1/10, you can configure port 1/5 as a member of an AppleTalk protocol VLAN, an IP protocol VLAN, and an IPX protocol VLAN, and so on.

If the router interface for IP is configured on physical ports, then routing occurs independent of the Spanning Tree Protocol (STP). However, if the router interfaces are defined for IP VLAN, they are virtual routing interfaces and are subject to the rules of STP.

If your backbone consists of virtual routing interfaces all within the same STP domain, it is a bridged backbone, not a routed one. This means that the set of backbone interfaces that are blocked by STP will be blocked for routed protocols as well. The routed protocols will be able to cross these paths only when the STP state of the link is FORWARDING. This problem is easily avoided by proper network design.

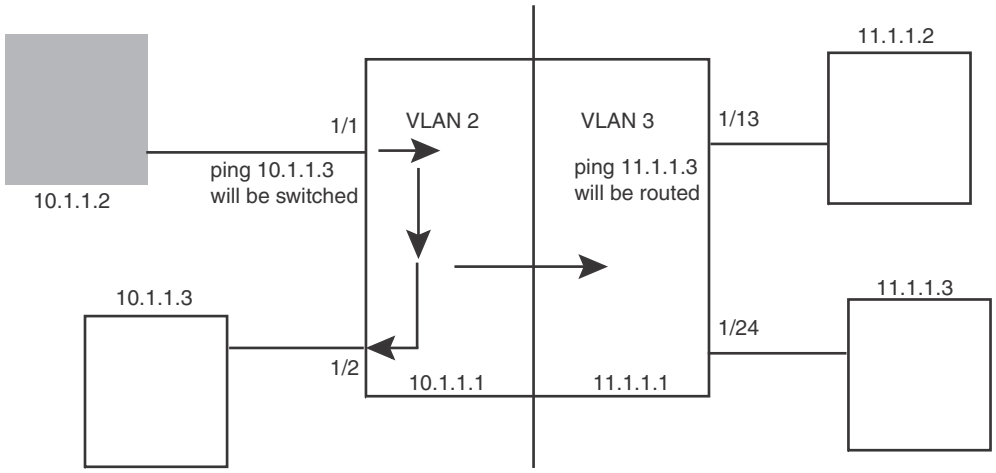
When designing an ISR network, pay attention to your use of virtual routing interfaces and the spanning-tree domain. If Layer 2 switching of your routed protocols (IP, IPX, AppleTalk) is not required across the backbone, then the use of virtual routing interfaces can be limited to edge switch ports within each router. Full backbone routing can be achieved by configuring routing on each physical interface that connects to the backbone. Routing is independent of STP when configured on a physical interface.

If your ISR design requires that you switch IP, IPX, or Appletalk at Layer 2 while simultaneously routing the IP protocol over a single backbone, then create multiple port-based VLANs and use VLAN tagging on the backbone links to separate your Layer 2 switched and Layer 3 routed networks.

There is a separate STP domain for each port-based VLAN. Routing occurs independently across port-based VLANs or STP domains. You can define each end of each backbone link as a separate tagged port-based VLAN. Routing will occur independently across the port-based VLANs. Because each port-based VLAN's STP domain is a single point-to-point backbone connection, you are guaranteed to never have an STP loop. STP will never block the virtual router interfaces within the tagged port-based VLAN, and you will have a fully routed backbone.

A PowerConnect router offers the ability to create a virtual routing interface within a Layer 2 STP port-based VLAN or within each IP protocol VLAN. This combination of multiple Layer 2 or Layer 3 broadcast domains and virtual routing interfaces are the basis for the very powerful Integrated Switch Routing (ISR) technology. ISR is very flexible and can solve many networking problems.

FIGURE 8 Example of two separate backbones for the same protocol



The following is a sample configuration for the illustration above.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# tagged e 1/1 to 1/2
NetIron(config-vlan-2)# router-inter ve 2
NetIron(config-vlan-2)# exit
NetIron(config)# vlan 3
NetIron(config-vlan-3)# tagged e 1/13 to 1/24
NetIron(config-vlan-3)# router-int ve 3
NetIron(config-vlan-3)# exit
NetIron(config)# interface ve 2
NetIron(config-ve-2)# ip address 10.1.1.1/24
NetIron(config-if-e1000-2/1)# exit
NetIron(config)# interface ve 3
NetIron(config-ve-3)# ip address 11.1.1.1/24
```

IP packets are bridged (switched) within the same protocol VLAN if they are on the same subnet; they are routed if they are on a different VLAN.

VLAN groups

To simplify VLAN configuration when you have many VLANs with the same configuration, you can configure *VLAN groups*. When you create a VLAN group, the VLAN parameters you configure for the group apply to all the VLANs within the group.

The VLAN group feature allows you to create multiple port-based VLANs with identical port members. Since the member ports are shared by all the VLANs within the group, you must add the ports as tagged ports. This feature not only simplifies VLAN configuration but also allows you to have a large number of identically configured VLANs in a startup configuration file on the device's flash memory module. Normally, a startup configuration file with a large number of VLANs might not fit on the flash memory module. By grouping the identically configured VLANs, you can conserve space in the startup configuration file so that it fits on the flash memory module.

NOTE
Depending on the size of the VLAN ID range you want to use for the VLAN group, you might need to allocate additional memory for VLANs. To allocate additional memory, refer to "Allocating memory for more VLANs or virtual routing interfaces" on page 260.

Configuring a VLAN group

To configure a VLAN group, perform the tasks listed below.

1. Create the VLAN group and assign the VLANs to that group.

```
NetIron(config)# vlan-group 1 vlan 2 to 1000
```

Syntax: **[no] vlan-group** <*num*> **vlan** <*vlan-id*> **to** <*vlan-id*>

The <*num*> parameter specifies the VLAN group ID.

The **vlan** <*vlan-id*> **to** <*vlan-id*> parameters specify a continuous range (with no gaps) of VLAN IDs that have not been configured in the CLI. Specify the low VLAN ID first and the high VLAN ID second. The command adds all the VLANs in the range to the VLAN group.

If a VLAN within the range you specify is already configured, the CLI does not add the group but instead displays an error message. If this happens, create the group by specifying a valid contiguous range that does not include the VLAN. Then add more VLANs to the group after the CLI changes to the configuration level for the group.

NOTE
The device's memory must be configured to contain at least the number of VLANs you specify for the higher end of the range. For example, if you specify 2048 as the VLAN ID at the high end of the range, you first must increase the memory allocation for VLANs to 2048 or higher. Refer to "Allocating memory for more VLANs or virtual routing interfaces" on page 260.

2. The CLI directs you to the VLAN group configuration level. Add tagged ports to the group. Since all the VLANs in the group share the ports, you must add the ports as tagged ports.

```
NetIron(config-vlan-group-1)# tagged e 1/1 to 1/2
```

Syntax: **[no] tagged ethernet [to** <*slot-number*>**/**<*port-number*>**> | ethernet** <*slot-number*>**/**<*port-number*>**]**

Using the **no tagged ethernet** command causes the following error message such as the following to appear.

```
NetIron(config-vlan-10)#no tagged ethernet 4/2
error - ports ether 4/1 to 4/2 are not tagged members of vlan 10
```

This message is normal and indicates that the configuration has take effect. It does not indicate that an error condition has occurred.

3. If required, you can add and remove individual VLANs or VLAN ranges from the VLAN group configuration level. For example, to add VLANs 1001 and 1002 to VLAN group 1 and remove VLANs 900 through 1000, enter the following commands.

```
NetIron(config-vlan-group-1)# add-vlan 1001 to 1002
NetIron(config-vlan-group-1)# remove-vlan 900 to 1000
```

Syntax: **[no] add-vlan** <*vlan-id*> **[to** <*vlan-id*>**]**

Syntax: **[no] remove-vlan** <*vlan-id*> **[to** <*vlan-id*>**]**

Verifying VLAN group configuration

To verify configuration of VLAN groups, display the running configuration file. If you have saved the configuration to the startup configuration file, you also can verify the configuration by displaying the startup configuration file. The following example shows the running configuration information for the VLAN group configured in the previous examples. The information appears in the same way in the startup configuration file.

```
NetIron(config)# show running-config
```

lines not related to the VLAN group omitted...

```
vlan-group 1 vlan 2 to 900
add-vlan 1001 to 1002
tagged ethernet 1/1 to 1/2
```

Displaying information about VLAN groups

To display VLAN group configuration information, enter the following command.

```
NetIron# show vlan-group 10

Configured VLAN-Group entries : 1
Maximum VLAN-Group entries : 32

VLAN-GROUP 10
Number of VLANs: 4
VLANs: 10 to 13
Tagged ports: ethernet 3/1
```

The example shows configuration information for two VLAN groups, group 1 and group 2.

Syntax: show vlan-group [<*group-id*>]

The <*group-id*> specifies a VLAN group. If you do not use this parameter, the configuration information for all the configured VLAN groups is displayed.

Configuring super aggregated VLANs

A super aggregated VLAN allows multiple VLANs to be placed within another VLAN. This feature allows you to construct Layer 2 paths and channels. A path contains multiple channels, each of which is a dedicated circuit between two end points. The two devices at the end points of the channel appear to each other to be directly attached. The network that connects them is transparent to the two devices.

You can aggregate up to 4090 VLANs within another VLAN. This provides a total VLAN capacity on one PowerConnect router of 16,728,100 channels (4090 * 4090).

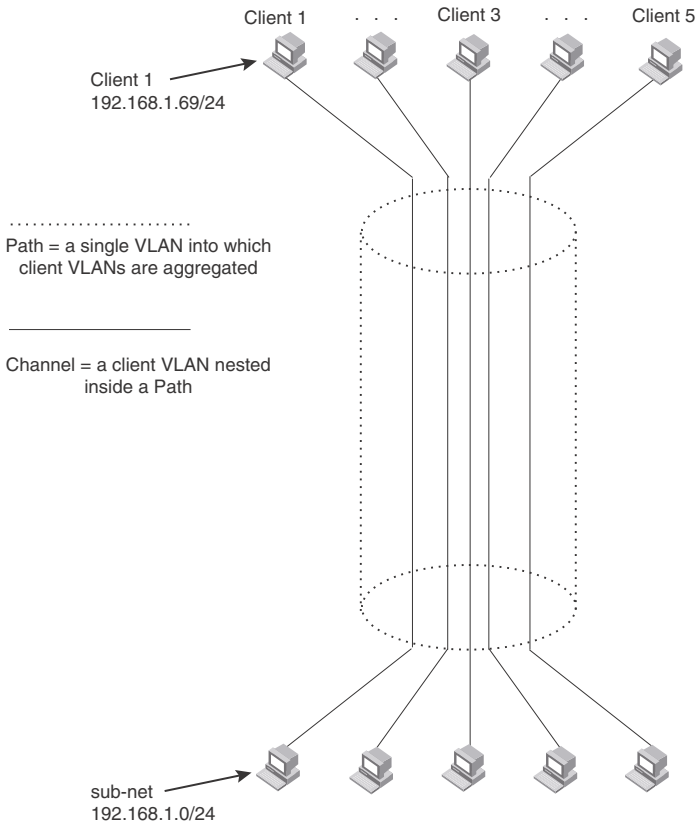
The devices connected through the channel are not visible to devices in other channels. Therefore, each client has a private link to the other side of the channel.

Super aggregated VLANs are useful for applications such as Virtual Private Network (VPN) or Transparent LAN Services (TLS) in which you need to provide a private, dedicated Ethernet connection to individual clients to transparently reach its subnet across multiple networks. The feature allows point-to-point and point-to-multipoint connections.

Figure 9 shows a conceptual picture of the service that aggregated VLANs provide.

In Super Aggregated VLANs, the outer VLAN (i.e. path) and the inner VLAN (i.e. channel) use different tag types. For example, the outer VLAN tag-type can be 9100 and the inner VLAN tag-type can be 8100 as shown in Figure 11.

FIGURE 9 Conceptual model of the super aggregated VLAN application



Each client connected to the edge device is in its own port-based VLAN. All the clients' VLANs are aggregated by the edge device into a single VLAN for connection to the core.

The device that aggregates the VLANs forwards the aggregated VLAN traffic through the core. The core can consist of multiple devices that forward the aggregated VLAN traffic. The edge device at the other end of the core separates the aggregated VLANs into the individual client VLANs before forwarding the traffic. The edge devices forward the individual client traffic to the clients. For the clients' perspective, the channel is a direct point-to-point link.

Figure 10 shows an example application that uses aggregated VLANs. This configuration includes the client connections shown in Figure 9.

FIGURE 10 Example super aggregated VLAN application

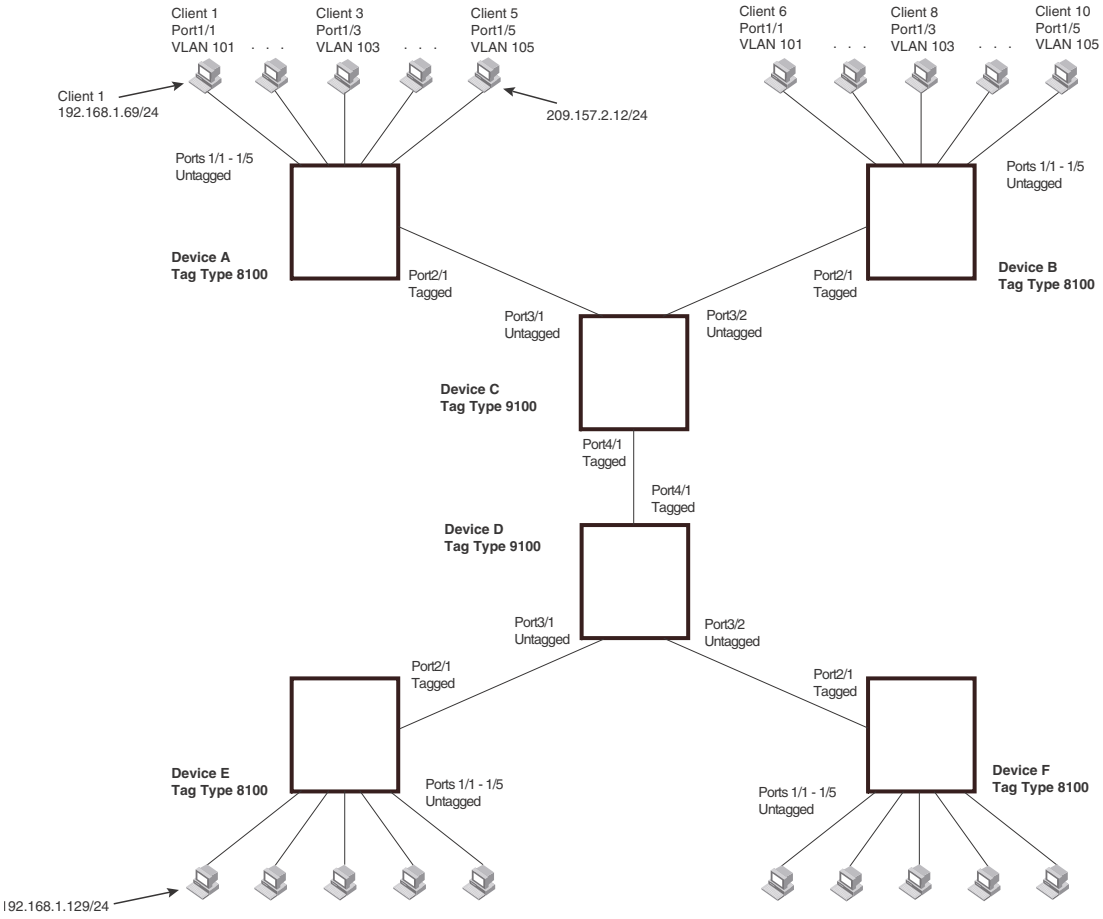


Figure 7 shows a collocation service provides private channels for multiple clients. Although the same devices are used for all the clients, the VLANs ensure that each client receives its own Layer 2 broadcast domain, separate from the broadcast domains of other clients. For example, client 1 cannot ping client 5.

The clients at each end of a channel appear to each other to be directly connected and thus can be on the same subnet and use network services that require connection to the same subnet. In this example, client 1 is in subnet 192.168.1.0/24 and so is the device at the other end of client 1's channel.

Since each VLAN configured on the core devices is an aggregate of multiple client VLANs, the aggregated VLANs greatly increase the number of clients a core device can accommodate.

This example shows a single link between the core devices. However, you can use a LAG group to add link-level redundancy.

Configuring aggregated VLANs

A maximum of 1526 bytes are supported on ports where super-aggregated VLANs are configured. This allows for an additional 8 bytes over the untagged port maximum to allow for support of two VLAN tags.

To configure aggregated VLANs, configure tagged and untagged VLANs on the edge device, then configure the aggregated and other VLANs on the core device. Perform the tasks listed below.

1. On each edge device, configure a separate port-based VLAN for each client connected to the edge device. In each client VLAN:

   - Add the port connected to the client as an untagged port.
   - Add the port connected to the core device (the device that will aggregate the VLANs) as a tagged port. This port must be tagged because all the client VLANs share the port as an uplink to the core device.

   For example, to configure device A in Figure 10 on page 250, enter commands such as the following.

```
NetIron(config)# vlan 101
NetIron(config-vlan-101)# tagged ethernet 2/1
NetIron(config-vlan-101)# untagged ethernet 1/1
NetIron(config-vlan-101)# exit
NetIron(config)# vlan 102
NetIron(config-vlan-102)# tagged ethernet 2/1
NetIron(config-vlan-102)# untagged ethernet 1/2
NetIron(config-vlan-102)# exit
NetIron(config)# vlan 103
NetIron(config-vlan-103)# tagged ethernet 2/1
NetIron(config-vlan-103)# untagged ethernet 1/3
NetIron(config-vlan-103)# exit
NetIron(config)# vlan 104
NetIron(config-vlan-104)# tagged ethernet 2/1
NetIron(config-vlan-104)# untagged ethernet 1/4
NetIron(config-vlan-104)# exit
NetIron(config)# vlan 105
NetIron(config-vlan-105)# tagged ethernet 2/1
NetIron(config-vlan-105)# untagged ethernet 1/5
NetIron(config-vlan-105)# exit
NetIron(config)# write memory
```

   Syntax: [no] **vlan** *<vlan-id>*

   Syntax: [no] **untagged | tagged ethernet** *<slot-number>*/*<port-number>* [**to** *<slot-number>*/*<port-number>* | **ethernet** *<slot-number>*/*<port-number>*]

   The **tagged** command adds the port that the device uses for the uplink to the core device.

   The **untagged** command adds the ports connected to the individual clients.

2. On each core device:

   - Configure a VLAN tag type (tag ID) that is different than the tag type used on the edge devices. If you use the default tag type (8100) on the edge devices, set the tag type on the core devices to another value, such as 9100. The tag type must be the same on all the core devices. The edge devices also must have the same tag type but the type must be different from the tag type on the core devices.

   NOTE
   You can enable the Spanning Tree Protocol (STP) on the edge devices or the core devices, but not both. If you enable STP on the edge devices and the core devices, STP will prevent client traffic from travelling through the core to the other side.

   For example, to configure the aggregated VLANs on device C in Figure 10 on page 250, enter the following commands.

```
NetIron(config)# tag-type 9100
NetIron(config)# vlan 101
NetIron(config-vlan-101)# tagged ethernet 4/1
NetIron(config-vlan-101)# untagged ethernet 3/1
NetIron(config-vlan-101)# exit
NetIron(config)# vlan 102
NetIron(config-vlan-102)# tagged ethernet 4/1
NetIron(config-vlan-102)# untagged ethernet 3/2
NetIron(config-vlan-102)# exit
NetIron(config)# write memory
```

Syntax: [no] **tag-type** <*num*> [**ethernet** <*slot/port*>]

The **num** variable is the hexadecimal ethernet tag type. Default value is 8100.

Complete CLI examples

The following sections show all the Aggregated VLAN configuration commands on the devices in Figure 10 on page 250.

NOTE
In these examples, the configurations of the edge devices (A, B, E, and F) are identical. The configurations of the core devices (C and D) also are identical. The aggregated VLAN configurations of the edge and core devices on one side must be symmetrical (in fact, a mirror image) to the configurations of the devices on the other side. For simplicity, the example in Figure 10 on page 250 is symmetrical in terms of the port numbers. This allows the configurations for both sides of the link to be the same. If your configuration does not use symmetrically arranged port numbers, the configurations should not be identical but must use the correct port numbers.

Commands for device A

```
NetIron-A(config)# vlan 101
NetIron-A(config-vlan-101)# tagged ethernet 2/1
NetIron-A(config-vlan-101)# untagged ethernet 1/1
NetIron-A(config-vlan-101)# exit
NetIron-A(config)# vlan 102
NetIron-A(config-vlan-102)# tagged ethernet 2/1
NetIron-A(config-vlan-102)# untagged ethernet 1/2
NetIron-A(config-vlan-102)# exit
NetIron-A(config)# vlan 103
NetIron-A(config-vlan-103)# tagged ethernet 2/1
NetIron-A(config-vlan-103)# untagged ethernet 1/3
NetIron-A(config-vlan-103)# exit
NetIron-A(config)# vlan 104
NetIron-A(config-vlan-104)# tagged ethernet 2/1
NetIron-A(config-vlan-104)# untagged ethernet 1/4
NetIron-A(config-vlan-104)# exit
NetIron-A(config)# vlan 105
NetIron-A(config-vlan-105)# tagged ethernet 2/1
NetIron-A(config-vlan-105)# untagged ethernet 1/5
NetIron-A(config-vlan-105)# exit
NetIron-A(config)# write memory
```

Commands for device B

The commands for configuring device B are identical to the commands for configuring device A. Notice that you can use the same channel VLAN numbers on each device. The devices that aggregate the VLANs into a path can distinguish between the identically named channel VLANs based on the ID of the path VLAN.

```
NetIron-B(config)# vlan 101
NetIron-B(config-vlan-101)# tagged ethernet 2/1
NetIron-B(config-vlan-101)# untagged ethernet 1/1
NetIron-B(config-vlan-101)# exit
NetIron-B(config)# vlan 102
NetIron-B(config-vlan-102)# tagged ethernet 2/1
NetIron-B(config-vlan-102)# untagged ethernet 1/2
NetIron-B(config-vlan-102)# exit
NetIron-B(config)# vlan 103
NetIron-B(config-vlan-103)# tagged ethernet 2/1
NetIron-B(config-vlan-103)# untagged ethernet 1/3
NetIron-B(config-vlan-103)# exit
NetIron-B(config)# vlan 104
NetIron-B(config-vlan-104)# tagged ethernet 2/1
NetIron-B(config-vlan-104)# untagged ethernet 1/4
NetIron-B(config-vlan-104)# exit
NetIron-B(config)# vlan 105
NetIron-B(config-vlan-105)# tagged ethernet 2/1
NetIron-B(config-vlan-105)# untagged ethernet 1/5
NetIron-B(config-vlan-105)# exit
NetIron-B(config)# write memory
```

Commands for device C

Since device C is aggregating channel VLANs from devices A and B into a single path, you need to change the tag type and enable VLAN aggregation.

```
NetIron-C(config)# tag-type 9100
NetIron-C(config)# vlan 101
NetIron-C(config-vlan-101)# tagged ethernet 4/1
NetIron-C(config-vlan-101)# untagged ethernet 3/1
NetIron-C(config-vlan-101)# exit
NetIron-C(config)# vlan 102
NetIron-C(config-vlan-102)# tagged ethernet 4/1
NetIron-C(config-vlan-102)# untagged ethernet 3/2
NetIron-C(config-vlan-102)# exit
NetIron-C(config)# write memory
```

Commands for device D

Device D is at the other end of path and separates the channels back into individual VLANs. The tag type must be the same as tag type configured on the other core device (Device C). In addition, VLAN aggregation also must be enabled.

```
NetIron-D(config)# tag-type 9100
NetIron-D(config)# vlan 101
NetIron-D(config-vlan-101)# tagged ethernet 4/1
NetIron-D(config-vlan-101)# untagged ethernet 3/1
NetIron-D(config-vlan-101)# exit
NetIron-D(config)# vlan 102
```

```
NetIron-D(config-vlan-102)# tagged ethernet 4/1
NetIron-D(config-vlan-102)# untagged ethernet 3/2
NetIron-D(config-vlan-102)# exit
NetIron-D(config)# write memory
```

Commands for device E

Since the configuration in Figure 10 on page 250 is symmetrical, the commands for configuring device E are identical to the commands for configuring device A.

```
NetIron-E(config)# vlan 101
NetIron-E(config-vlan-101)# tagged ethernet 2/1
NetIron-E(config-vlan-101)# untagged ethernet 1/1
NetIron-E(config-vlan-101)# exit
NetIron-E(config)# vlan 102
NetIron-E(config-vlan-102)# tagged ethernet 2/1
NetIron-E(config-vlan-102)# untagged ethernet 1/2
NetIron-E(config-vlan-102)# exit
NetIron-E(config)# vlan 103
NetIron-E(config-vlan-103)# tagged ethernet 2/1
NetIron-E(config-vlan-103)# untagged ethernet 1/3
NetIron-E(config-vlan-103)# exit
NetIron-E(config)# vlan 104
NetIron-E(config-vlan-104)# tagged ethernet 2/1
NetIron-E(config-vlan-104)# untagged ethernet 1/4
NetIron-E(config-vlan-104)# exit
NetIron-E(config)# vlan 105
NetIron-E(config-vlan-105)# tagged ethernet 2/1
NetIron-E(config-vlan-105)# untagged ethernet 1/5
NetIron-E(config-vlan-105)# exit
NetIron-E(config)# write memory
```

Commands for device F

The commands for configuring device F are identical to the commands for configuring device E. In this example, since the port numbers on each side of the configuration in Figure 10 on page 250 are symmetrical, the configuration of device F is also identical to the configuration of device A and device B.

```
NetIron-F(config)# vlan 101
NetIron-F(config-vlan-101)# tagged ethernet 2/1
NetIron-F(config-vlan-101)# untagged ethernet 1/1
NetIron-F(config-vlan-101)# exit
NetIron-F(config)# vlan 102
NetIron-F(config-vlan-102)# tagged ethernet 2/1
NetIron-F(config-vlan-102)# untagged ethernet 1/2
NetIron-F(config-vlan-102)# exit
NetIron-F(config)# vlan 103
NetIron-F(config-vlan-103)# tagged ethernet 2/1
NetIron-F(config-vlan-103)# untagged ethernet 1/3
NetIron-F(config-vlan-103)# exit
NetIron-F(config)# vlan 104
NetIron-F(config-vlan-104)# tagged ethernet 2/1
NetIron-F(config-vlan-104)# untagged ethernet 1/4
NetIron-F(config-vlan-104)# exit
NetIron-F(config)# vlan 105
```

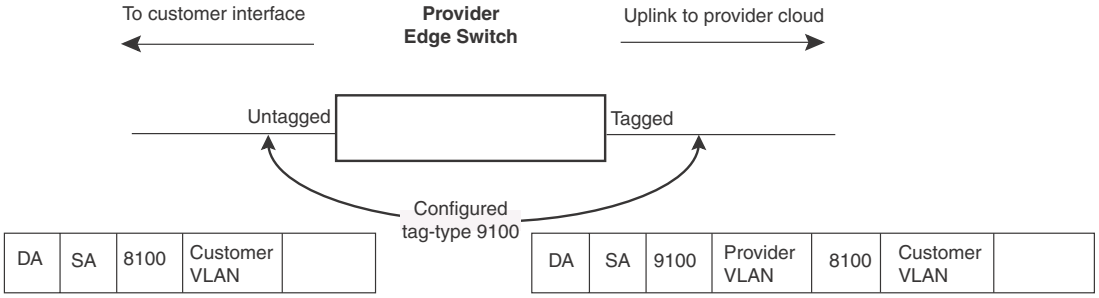
```
NetIron-F(config-vlan-105)# tagged ethernet 2/1
NetIron-F(config-vlan-105)# untagged ethernet 1/5
NetIron-F(config-vlan-105)# exit
NetIron-F(config)# write memory
```

Configuring 802.1q-in-q tagging

802.1Q-in-Q tagging enables you to configure 802.1Q tag-types on a group of ports, such as LAG ports, thereby enabling the creation of two identical 802.1Q tags (802.1Q-in-Q tagging) on a single device. This feature improves SAV interoperablilty between Dell devices and other vendors' devices that support the 802.1Q tag-types, but are not very flexible with the tag-types they accept.

Figure 11 shows an 802.1Q configuration example.

FIGURE 11 SAV configuration example

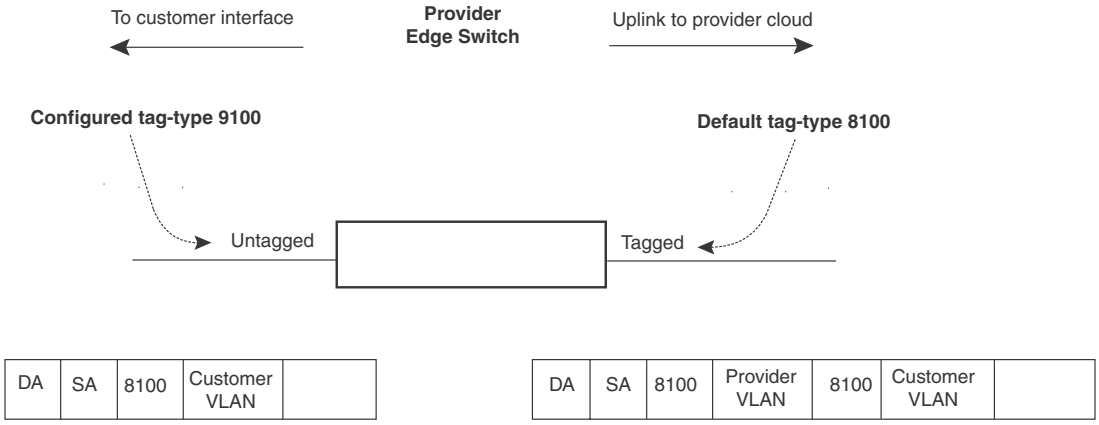


As shown in Figure 11, the ports to customer interfaces are untagged, whereas the uplink ports to the provider cloud are tagged, because multiple client VLANs share the uplink to the provider cloud. In this example, the PowerConnect router treats the customer's private VLAN ID and 8100 tag type as normal payload, and adds the 9100 tag type to the packet when the packet is sent to the uplink and forwarded along the provider cloud.

As long as the switches in the provider's network support the 9100 tag type, the data gets switched along the network. However, devices that do not support the 9100 tag type may not properly handle the packets.

Figure 12 and Figure 13 show an example application of 802.1Q-in-Q.

FIGURE 12 802.1Q-in-Q configuration example



In Figure 14, the untagged ports (to customer interfaces) accept frames that have any 802.1Q tag other than the configured tag-type 9100. These packets are considered untagged on this incoming port and are re-tagged when they are sent out of the uplink towards the provider. The 802.1Q tag-type on the uplink port is 8100, so the PowerConnect router will switch the frames to the uplink device with an additional 8100 tag, thereby supporting devices that only support this method of VLAN tagging.

Configuration rules

Follow the rules below when configuring 802.1q-in-q tagging:

- The PowerConnect router supports per port tag-type configuration. Consequently, each port can have its own tag-type setting.
- The default tag-type for a port is 8100
- The PowerConnect router supports 802.1q-in-q tagging where the inner and outer tag can have different or same tag-type values. This feature maximizes interoperablilty with third-party devices.

Enabling 802.1Q-in-Q tagging

To enable the 802.1Q-in-Q feature, configure an 802.1Q tag type on the untagged edge links (the customer ports) to any value other than the 802.1Q tag for incoming traffic.

For example, in Figure 13, the 802.1Q tag on the untagged edge links (ports 11 and 12) is 9100, whereas, the 802.1Q tag for incoming traffic is 8100.

To configure 802.1 Q-in-Q tagging as shown in Figure 13, enter commands such as the following on the untagged edge links of devices C and D.

```
NetIron(config)# tag-type 9100 e 3/1 to 3/2
```

Syntax: **[no] tag-type** <*num*> **[ethernet** <*slot-number*>/<*port-number*> **[to** <*slot-number*>/<*port-number*>**]]**

The <*num*> parameter specifies the tag-type number and can be a hexadecimal value from 0 - ffff. The default is 8100.

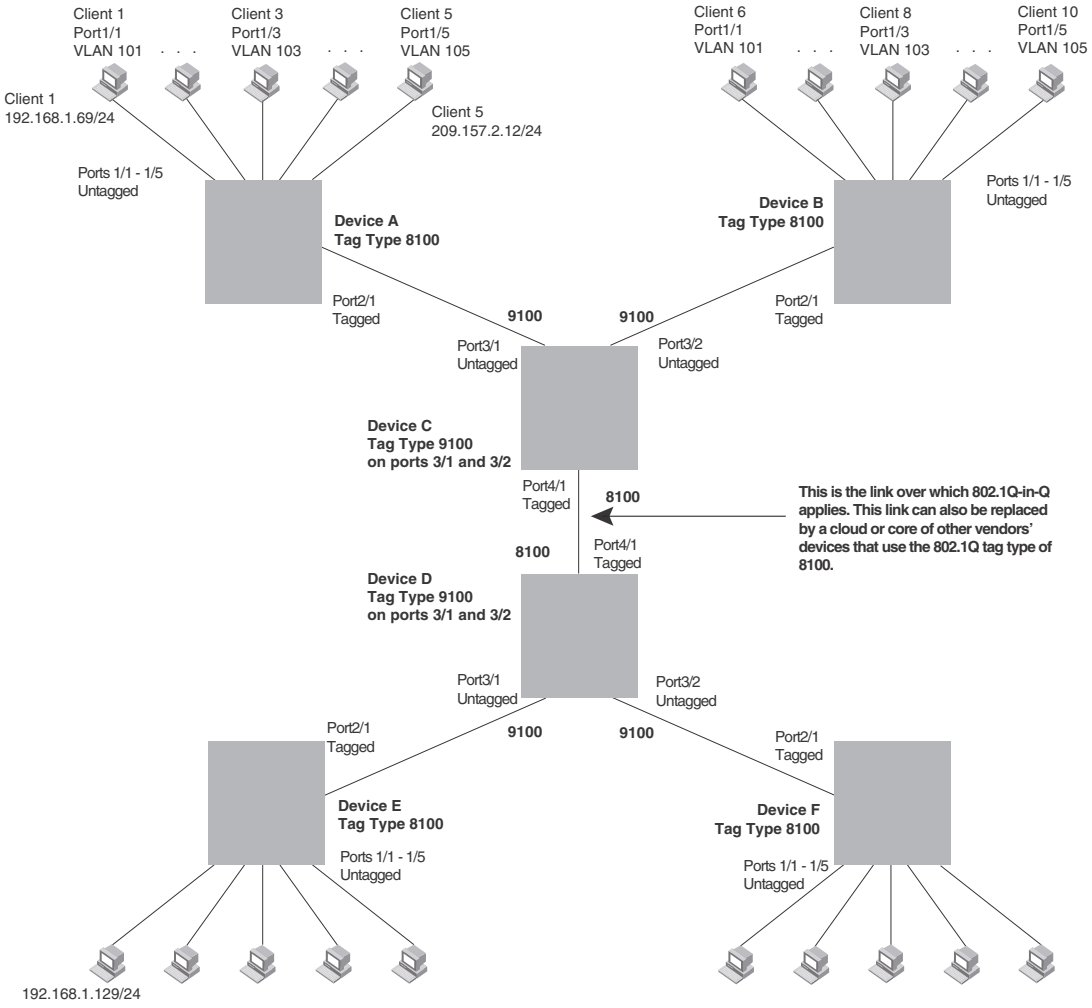
The **ethernet** <*port number*> **to** <*port number*> parameter specifies the ports that will use the defined 802.1Q tag. This parameter operates with the following rules:

- If you do not specify a port or range of ports, the 802.1Q tag applies to all Ethernet ports on the device.

Example configuration

Figure 13 shows an example 802.1Q-in-Q configuration.

FIGURE 13 Example 802.1Q-in-Q configuration



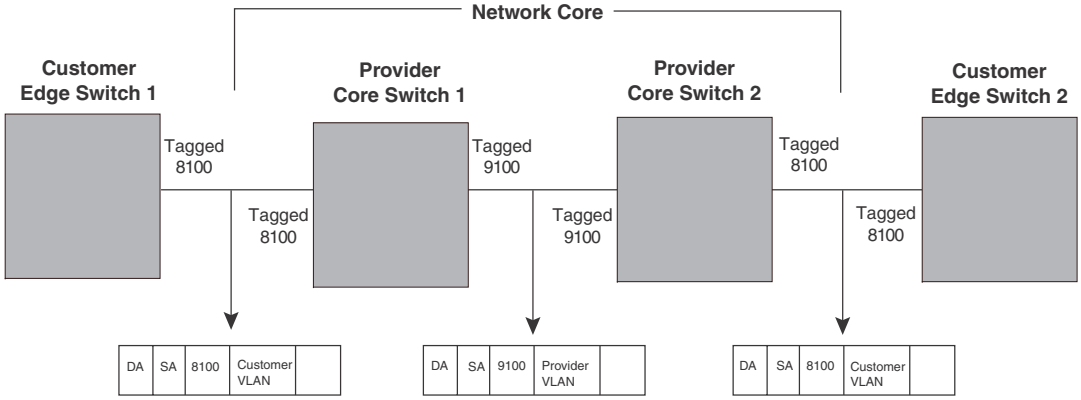
Configuring 802.1q tag-type translation

The introduction of 802.1q tag-type translation provides finer granularity for configuring multiple 802.1q tag-types on a single device, by enabling you to configure 802.1q tag-type per port. This enhancement allows for tag-type translation from one port to the next on tagged interfaces.

802.1Q tag-type translation enables you to configure a separate 802.1q tag-type per port, allowing for tag-type translation from one port to the next on tagged interfaces.

Figure 14 shows a basic example application of the 802.1q tag-type translation feature.

FIGURE 14 802.1q Tag-type translation configuration example 1



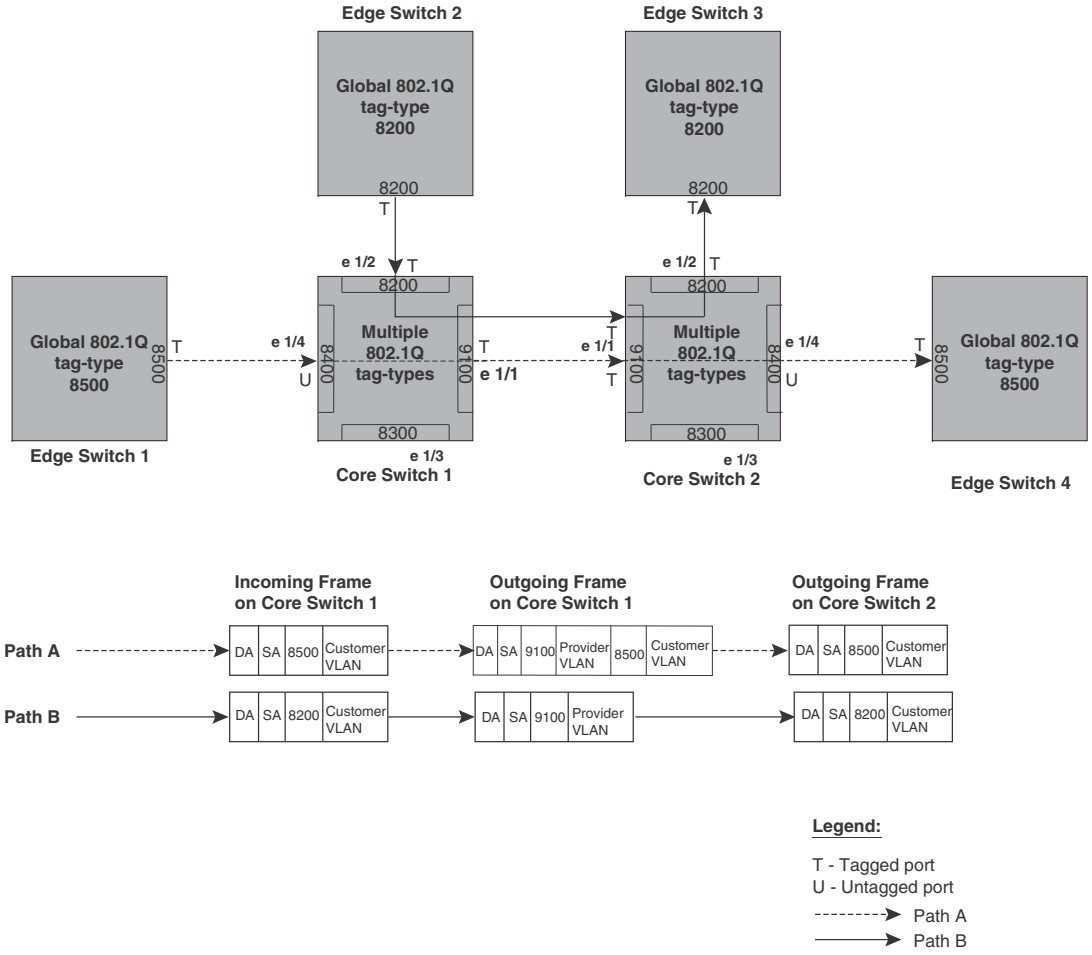
As illustrated in Figure 14, the devices process the packet as follows:

- Customer Edge Switch 1 sends a packet with an 802.1q tag-type of 8100 to Provider Core Switch 1.
- Since the customer-facing interface on Provider Core Switch 1 has the same 802.1q tag-type as the incoming packet, it removes the 8100 tag-type and replaces (translates) it with the 9100 tag-type as it sends the packet to the uplink (Provider Core Switch 2).
- The same process occurs between Provider Core Switch 2 and Customer Edge Switch 2.

Figure 14 shows a simple application of the 802.1q tag-type translation in which all of the ports are tagged and the tag-types between devices match. In this example, each device performs the 802.1q tag-type translation as the packet traverses the network.

Figure 15 shows a more complex example application in which some ports are untagged, not all tag-types between devices match, and the core devices have multiple tag-types. In this example, the tag-type translation feature integrates packets that have single and double tag-types.

FIGURE 15 802.1q Tag-type translation configuration example 2



As illustrated in Figure 15, the devices process the packets as follows:

- Path A: When Core Switch 1 receives the tagged packet from Edge Switch 1, it keeps the 8500 tag-type in the frame header (because the incoming port on Core Switch 1 is untagged) and adds the 9100 tag-type as it sends the packet to the uplink (Core Switch 2). In this case, the packet is double-tagged as it travels between the core devices.
- Path B: When Core Switch 1 receives the tagged packet from Edge Switch 2, it removes the 8200 tag-type and replaces (translates) it with the 9100 tag-type as it sends the packet to the uplink (Core Switch 2).

Configuration rules

Configuration of tag-type ports on the PowerConnect router are on a per-port basis and follow the same rules as described in "Configuration rules" on page 256. Enabling 802.1q Tag-type Translation

To enable 802.1q tag-type translation, configure an 802.1q tag-type on the provider core link, between the provider core switches (refer to Figure 15). Enter commands such as the following.

```
NetIron(config)# tag-type 9100 e 1/1
NetIron(config)# tag-type 8200 e 1/2
NetIron(config)# tag-type 8300 e 1/3
NetIron(config)# tag-type 8400 e 1/4
```

Syntax: [no] tag-type <*num*> **[ethernet** <*slot-number*>/<*port-number*> **[to** <*slot-number*>/<*port-number*>**]]**

The <*num*> parameter specifies the tag-type number and can be a hexadecimal value from 0 - ffff. The default is 8100.

The <*slot-number*>/<*port-number*> [**to** <*slot-number*>/<*port-number*>] parameter specifies the ports that will use the defined 802.1q tag-type. This parameter operates with the following rules:

- If the port that you specify is part of a multi-slot LAG, the device automatically applies the 802.1q tag-type to all of the ports that are part of the multi-slot LAG.
- If you do not specify a port or range of ports, the 802.1q tag-type applies to all Ethernet ports on the device.

Miscellaneous VLAN features

Allocating memory for more VLANs or virtual routing interfaces

By default, you can configure up to 512 VLANs and virtual routing interfaces on the router. Although this is the default maximum, the PowerConnect router can support up to 4090 VLANs and 4090 virtual routing interfaces.

NOTE
If many of your VLANs will have an identical configuration, you might want to configure VLAN groups.

If you need to configure more than 512 VLANs, enter commands such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# system-max vlan 2048
NetIron(config)# write memory
NetIron(config)# end
NetIron# reload
```

Syntax: [no] system-max vlan <*num*>

The <*num*> parameter specifies the maximum number of VLANs that can be configured. The minimum, maximum and default values for this parameter are described in Table 15.

NOTE
You must reload the system for the new parameters to take effect.

Configuring uplink ports within a port-based VLAN

You can configure a subset of the ports in a port-based VLAN as uplink ports. When you configure uplink ports in a port-based VLAN, the device sends all broadcast and unknown-unicast traffic from a port in the VLAN to the uplink ports, but not to other ports within the VLAN. Thus, the uplink ports provide tighter broadcast control within the VLAN.

For example, if two ports within a port-based VLAN are Gigabit ports attached to the network and the other ports in the VLAN are 10/100 ports attached to clients, you can configure the two ports attached to the network as uplink ports. In this configuration, broadcast and unknown-unicast traffic in the VLAN does not go to all ports in the VLAN. The traffic goes only to the uplink ports. The clients on the network do not receive broadcast and unknown-unicast traffic from other ports, including other clients.

To configure a port-based VLAN containing uplink ports, enter commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# untag ethernet 1/1 to 1/20
NetIron(config-vlan-10)# untag ethernet 2/1 to 2/2
NetIron(config-vlan-10)# uplink-switch ethernet 2/1 to 2/2
```

Syntax: [no] uplink-switch ethernet <*port-number*> **[to** <*port-number*> **| ethernet** <*port-number*>**]**

In this example, ports 1 - 20 on slot 1 and ports 1 - 2 on slot 2 are added to port-based VLAN 10. The two ports on slot 2 are then configured as uplink ports.

Configuring control protocols in VLANs

You can configure the following protocols on a VLAN:

- Foundry MRP (Refer to 13, “Metro Ring Protocol”.)
- VSRP (Refer to 15, “Virtual Switch Redundancy Protocol (VSRP)”.)
- STP (Refer to 11, “Configuring Spanning Tree Protocol”.)
- RSTP (Refer to 12, “Configuring Rapid Spanning Tree Protocol”.)

Hardware flooding for layer 2 multicast and broadcast packets

Broadcast and multicast packets do not have a specific recipient. In order for these "special" packets to reach their intended recipient, they need to be sent on all ports of the VLAN (or "flooded" across the VLAN).

You must enable hardware flooding for Layer 2 multicast and broadcast packets on the PowerConnect router . (Layer 2 multicast packets have a multicast address in the destination MAC address field.)

You can enable hardware flooding for Layer 2 multicast and broadcast packets on a per-VLAN basis.

Example

```
NetIron(config)#
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast-flooding
NetIron(config-vlan-2)# exit
```

Syntax: [no] multicast-flooding

NOTES:

- This feature cannot be enabled on an empty VLAN; the VLAN must already have ports assigned to it prior to enabling this feature.
- This feature is not supported on Layer 3 protocol-based VLANs.
- If you enable this feature on a VLAN that includes a LAG group, hardware flooding for Layer 2 multicast and broadcast packets occurs only on the LAG group's primary port. Multicast and broadcast traffic for the other ports in the LAG group is handled by software.

Unknown unicast flooding on VLAN ports

Unknown unicast packets do not have a specific (or unicast) recipient. In order for these "special" packets to reach their intended recipient, they needed to be sent on all ports of the VLAN (or "flooded" across the VLAN).

You must enable *hardware* flooding for unknown unicast packets on the PowerConnect router. It is disabled by default.

To enable unicast hardware flooding on a VLAN ports and enable software flooding, enter commands such as the following.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# unknown-unicast-flooding
NetIron(config-vlan-2)# exit
```

Syntax: [no] unknown-unicast-flooding

Configuring VLAN CPU protection

VLAN CPU protection is recommended for the VLANs which are intended for pure Layer2 use. This feature will protect the CPU from the flooding of unknown-unicast or multicast or broadcast L2 packets on that VLAN. When using routing protocols (like OSPF etc.) on a specific VLAN, you need to disable VLAN CPU protection for it to work. This feature is intended for Layer2 applications and not for Layer3 routing applications.

VLAN CPU protection is enabled per VLAN. To enable VLAN CPU protection on a VLAN, enter the following command.

```
NetIron(config)# vlan 24
NetIron(config-vlan-24)# vlan-cpu-protection
```

Syntax: [no] vlan-cpu-protection

NOTE
The Multi-Service IronWare software has been enhanced to support the dynamic growth of the protocol and flooding sub-partitions in the L2 CAM to grant them a higher priority. Because of this enhancement, the **system-max hw-flooding** command is no longer required and has been retired from the software.

Command changes to support 8x10G modules

The following commands changed to support NI-MLX-10x8G modules.

Deprecated commands:

- The **vlan-counter exclude-overhead** command has been deprecated. The new command is the **statistics -> exclude-ethernet-overhead** command.
- The **byte-accounting** command under a VLAN has been has been deprecated on the NetIron MLX platforms and replaced by the **vlan-accounting on | off** command. In addition, a new global **vlan-policy -> vlan-accounting** command has been introduced to enable/disable accounting for all VLANs.
- The **clear vlan byte-accounting all-vlans** command has been deprecated. The new command is the **clear vlan all-vlan statistics** command.
- The **clear vlan byte-accounting** command has been has been deprecated. The new command is the **clear vlan statistics** command.

Existing display command:

- New options have been added to the **show vlan** command to display VLAN counters for the 8x10G module.

Deprecated commands

vlan-counter exclude-overhead

The **vlan-counter exclude-overhead** command has been deprecated. The new command is the **exclude-ethernet-overhead** command.

NOTE
The **statistics -> exclude-ethernet-overhead** command will replace the **vlan-counter exclude-overhead** command when upgrading to the new image.

By default, the VLAN byte counters include the 20-byte Ethernet overhead. You can use the **exclude-ethernet-overhead** command to direct the PowerConnect router to exclude this overhead when it counts the bytes, as shown in the example below.

```
PowerConnect(config-statistics)#exclude-ethernet-overhead
```

Syntax: [no] exclude-ethernet-overhead

To disable the configuration, use the **no exclude-ethernet-overhead** command.

byte-accounting

NOTE
The **byte-accounting** command has been replaced by the **vlan-accounting on|off** command at the VLAN configuration level for the NetIron MLX platform.

The **byte-accounting** command under a VLAN has been has been deprecated for the NetIron MLX. The new command for those platforms is the **vlan-accounting on l off** command. In addition a new global **vlan-policy -> vlan-accounting** command has also been introduced to enable/disable accounting for all VLANs.

NOTE
The **vlan-accounting on** command will replace the **byte-accounting** command under a VLAN when upgrading to the new image on the NetIron MLX platform.

The **vlan-accounting on | off** command at the VLAN level takes precedence over global configuration. For example, if VLAN accounting is globally enabled, and the user disables VLAN accounting on VLAN 10, then VLAN accounting for VLAN 10 is disabled.

By default, L2 VLAN accounting is globally enabled for all VLANs. The VLAN counters are polled every 50 seconds.

To disable VLAN accounting globally for all VLANs, enter the following command at the config-vlan-policy level of the CLI.

```
NetIron(config-vlan-policy)#no vlan-accounting
```

Syntax: [**no**] **vlan-accounting**

To disable VLAN accounting globally, enter the **no vlan-accounting** command.

To configure VLAN accounting for specific VLAN, enter the following command.

```
PowerConnect(config-vlan-10)# vlan-accounting on.
```

Syntax: **[no] vlan-accounting <**on **|** off**>**

The **vlan-accounting on** command enables counters for a specific VLAN. The **vlan-accounting off** command disables counters for a specific VLAN.

clear vlan all-vlans statistics

The **clear vlan byte-accounting all-vlans** command has been deprecated. The new command is the **clear vlan all-vlans statistics** command.

To clear VLAN counters for all VLANs, enter the following command.

```
NetIron# clear vlan all-vlans statistics
```

Syntax: **clear vlan all-vlans statistics**

clear vlan byte-accounting

The **clear vlan byte-accounting** command has been has been deprecated. The new command is the **clear vlan statistics** command.

To clear the VLAN counters on a specific VLAN, say VLAN 10, enter the following command.

```
NetIron# clear vlan 10 statistics
```

Syntax: **clear vlan** <vlan-id> **statistics**

Use the <vlan-id> parameter to specify the name of the VLAN to clear statistic counter on.

Existing display command:

The byte counter displayed by the output of **show vlan** command is the number of received bytes across all ports (both G2 and non-G2 ports) in the specified VLAN.

```
PowerConnect# show vlan
Configured PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 4090
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
L2 protocols : NONE
Untagged Ports : ethernet 2/1 to 2/20 ethernet 3/1 to 3/20 ethernet
PORT-VLAN 2, Name [None], Priority Level0
L2 protocols : NONE
ip-protocol VLAN, Dynamic port disabled
Name: basic
PORT-VLAN 1001, Name [None], Priority Level0
L2 protocols : MRP
Tagged Ports : ethernet 3/1 ethernet 3/12 to 3/13 ethernet 3/20
Bytes received : 6000
```

TABLE 41 show vlan output details

Configured PORT-VLAN entries	Number of port-based VLANs in the configuration.
Maximum PORT-VLAN entries: 4090	Maximum number of port-based VLANs that you can configure.
Default PORT-VLAN id	ID of the default VLAN.
PORT-VLAN	ID of the port-based VLAN.
Name	Name of the port-based VLAN. [None] appears if a name has not been assigned.
Priority Level	Priority level assigned to the port-based VLAN.
L2 protocols	Layer 2 control protocol configured on the VLAN.
Untagged or Tagged Ports	ID of the untagged or tagged ports that are members of the VLAN.
(protocol-based VLANs)	If protocol based VLANs are configured, their type and name appear after the list of ports.
Bytes received	Displays the number of received bytes across all ports in the specified VLAN.

Extended VLAN counters for 8x10G modules

The NI-MLX-10x8G module supports VLAN counters on the ingress and egress ports. The NI-MLX-10x8G module supports packet and byte accounting for 32K counters on both inbound and outbound traffic on a per-VLAN, per-port, per-priority basis. The 8x10G module supports 64 bit VLAN counters for both packet and byte accounting.

To support extended VLAN accounting, the following modes allow you to configure packet and byte accounting on a 8x10G module.

Priority mode - This mode allows accounting to be performed on a per VLAN, per port, per-priority basis. Priority mode is configured on per-module basis. By default, the per-priority accounting mode is disabled **i.e. accounting is done per VLAN, per port**..

Switched or routed separate mode - This mode allows you to specify whether the switched packet and routed packets should be counted separately or not. This mode is configured globally, and by default, switched packets and routed packets are counted together.

Refer to Table 42 on the number of unique port, VLAN's supported per PPCR based on the configuration of "Priority mode" and "Switched or routed separate mode"...

TABLE 42 Internal priority of switched and routed packets

Switched and routed packets	Account based on the internal priority of the packet- Yes or No	Number of unique port-VLANs that have counters (per-PPCR).
Switch or Route separately	Yes	2047 on ingress and 2047 on egress; each set having 16 counters
Switch or Route separately	No	16383 in ingress and 16383 on egress; each set having 2 counters
Switch or route combined	Yes	4095 on ingress and 4095 on egress; each set having 8 counters
Switch or route combined	No	32767 on ingress and 32767 on egress; each set having 1 counter

Configuring extended VLAN counters

The 8x10g modules supports the following global configuration commands.

Enabling accounting on per-slot basis

You can enable or disable per-VLAN/port,/priority accounting mode on all or a per-slot basis on the ingress and egress counters. Layer 2 VLAN accounting is enabled by default. and counters are polled once every 50 seconds. To enable accounting on per-slot basis, enter the following command.

Layer 2 VLAN accounting is enabled by default. Counters are polled once every 50 seconds

```
PowerConnect(config)#statistics
PowerConnect(config-statistics)#extended-counters priority all
```

Syntax: [no] extended-counters priority <all> **|** <*slot-number*>

The <*slot-number*> variable specifies the ID of 8x10 module on which you can perform accounting on per-slot basis.

If the <all> option is specified, the configuration command is remembered in the system when the 8x10 module is removed.

If you dynamically enable or disable the **extended-counters priority** configuration, the sum of counters displayed on a per-priority basis will not be same as the aggregate count displayed on a per-port or per-VLAN basis.

Enabling accounting on switched or routed packets

To enable or disable accounting on switched packets and routed packets separately, enter the following example:

```
PowerConnect(config)#statistics
PowerConnect(config-statistics)#extended-counters routed-switched
```

Syntax: [no] extended-counters routed-switched

If you dynamically enable or disable the **extended-counters routed-switched** configuration, the current counters are saved and added to the count of aggregate packet and byte counters on a per-port or per-VLAN basis and displayed in the output of combined counters.

Displaying VLAN counters

The **show vlan** commands changed to display port-vlan counters for 8x10G modules.

To display VLAN counters information for specific VLAN, enter the following command.

```
NetIron# show vlan 10 statistics
VLAN 10: Extended Routed/Switched Counters (only applicable for G2 modules):

Slot 12: < -- module with per-VLAN/port/priority based accounting
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                           <snip>
     p6   0                0                0                0
     p7   0                0                0                0
eth 12/2  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                           <snip>
     p6   0                0                0                0
     p7   0                0                0                0
eth 12/3         -- Extended-counter resource allocation failed –   < -- On
encountering Stats ID allocation failure
eth 12/4  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
<snip>
     p6   0                0                0                0
     p7   0                0                0                0

Slot 14: < -- module with per-VLAN/port based accounting
```

Syntax: show vlan <vlan id> **statistics** [detail |routed|switched]

The <vlan-id> parameter specifies the VLAN ID of the port.

The <slot/port> parameter specifies the interface module location of a 8x10g module.

Use the **routed** option to view the counters of routed packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **switched** option to view the counters of switched packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **detail** option to view the counters of routed packets, counters of switched packets, and counters of both routed and switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

If the “routed” or “switched” option is specified, counters for only routed and switched packets are displayed respectively, otherwise the output displays combined statistics for both routed and switched packets.

TABLE 43 Output descriptions of the **show vlan** command

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying VLAN counters for a specific port

To display VLAN counters information for specific port on a VLAN, enter the following command.

```
PowerConnect# show vlan 10 statistics ethernet 14/1
VLAN 10: Extended Routed/Switched Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
```

To display VLAN counters information for routed packets on a specific port on a VLAN, enter the following command.

```
PowerConnect# show vlan 10 statistics ethernet 14/1 routed
VLAN 10: Extended Routed Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
```

To display VLAN counters information for switched packets on a specific port on a VLAN, enter the following command.

```
PowerConnect# show vlan 10 statistics ethernet 14/1 switched
VLAN 10: Extended Switched Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
```

To display detailed VLAN counters information on a specific port on a VLAN, enter the following command.

```
PowerConnect# show vlan 10 statistics ethernet 14/1 detail
VLAN 10: Extended Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1
 Routed   0                0                0                0
 Switched 0                0                0                0
 Combined 0                0                0                0
```

Syntax: show vlan <vlan id**> statistics ethernet** <port-id> **[detail |routed |switched]**

The <vlan-id> parameter specifics the VLAN ID of the port.

The <slot/port> parameter specifies the interface module location of a 8x10g module.

Use the **routed** option to view the counters of routed packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **switched** option to view the counters of switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **detail** option to view the counters of routed packets, counters of switched packets, and counters of both routed and switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

If the “routed” or “switched” option is specified, counters for only routed and switched packets are displayed respectively, otherwise the output displays combined statistics for both routed and switched packets.

TABLE 44 Output description for the **show vlan** command

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Clearing extended VLAN counters

You use the following commands to clear the VLAN counters.

Clearing counters for all VLANs

To clear the ingress and egress packet and byte counters for routed packets and switched packets on all VLANs, enter the following command.

```
PowerConnect# clear vlan all-vlans statistics
```

Syntax: clear vlan all-vlans statistics [switched]

Enter the **switched** keyword to clear only the counters of switched packets. If you do not specify the **switched** keyword, the counters for both routed packets and switched packets are cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing counters for a specific VLAN

To clear the VLAN counters for a specific VLAN, enter the following command.

```
PowerConnect# clear vlan 10 statistics
```

Syntax: clear vlan <*vlan-id*> **statistics [switched]**

Use the <*vlan-id*> option to specify the VLAN ID of the port for which you want to clear the counters.

Enter the **switched** keyword to clear only the counters of switched packets. If you do not specify the **switched** keyword, the counters for both routed packets and switched packets are cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing VLAN and port counters

To clear VLAN, port, and priority counters for specific VLAN and port combinations, enter the following command.

```
PowerConnect# clear vlan 10 statistics ethernet 1/2 switched
```

Syntax: clear vlan <*vlan-id*> **statistics ethernet** <*port-id*> **[switched]**

Use the <*vlan-id*> option to specify the VLAN ID of the port for which you want to clear the counters.

Use the <*port-id*> option to specify the port for with you want to clear the counters.

Specify the **switched** keyword to clear counters for the switched packets. If the **switched** keyword is not specified the counters for both routed packets and switched packets will be cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing VLAN counters on a port with a specific priority

To clear counters for a specific port in a VLAN with specific priority, enter the following command.

```
PowerConnect# clear vlan 10 statistics ethernet 1/2 priority 3 switched
```

Syntax: clear vlan <*vlan-id*> **statistics ethernet** <*port-id*> **priority** <*0-7*> **[switched]**

Use the <*vlan-id*> option to specify the VLAN ID of the port for which you want to clear the counters.

Use the <*port-id*> option to specify the port for with you want to clear the counters.

Specify the **switched** keyword to clear counters for the switched packets. If the **switched** keyword is not specified the counters for both the routed packets and switched packets will be cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing extended counters statistics on a port

To clear all extended counters statistics simultaneously for a single port, enter the following command.

```
PowerConnect# clear statistics ethernet 1/2 extended counters
```

Syntax: clear statistics ethernet <*port_id or range* > **extended counters**

Use the <*port_id or range*> option to specify the port or range for which you want to clear the extended counters.

Clearing extended counters statistics on specific slot

To clear all extended counter statistics simultaneously for a slot, enter the following command.

```
PowerConnect# clear statistics slot 2 extended counters
```

Syntax: clear statistics slot <slot -id> extended counters

Use the <*slot-id*> option to specify the slot number for which you want to clear the extended counters.

IP interface commands

You can display and clear the counter details of the physical and virtual IP interfaces.

Displaying IP interface counters

You can display aggregate count of the routed packets and switched packets of an IP interface using the following command.

```
<< If Routed/Switched separate mode >>
NetIron# show ip interface ve 10 statistics
Extended Routed Counters (only applicable for G2 modules):
Total     RxPkts           TxPkts           RxBytes          TxBytes
          0                0                0                0

<< If Routed/Switched combined mode>>
NetIron# show ip interface ve 10 statistics
Extended Routed/Switched Counters (only applicable for G2 modules):
Total     RxPkts           TxPkts           RxBytes          TxBytes
                    0                0                0                0
```

Syntax: show ip interface ethernet <*port-id*> **statistics**

Specify the< *port id*> of the interface for which you want to display the routed and switched packets aggregate count.

TABLE 45 show ip interface command output details

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying IP virtual interface counters

To display the counters for each physical port of a virtual IP interface, use the following command.

```
NetIron#show ip interface ve 10 statistics ethernet 12/1

Extended Routed Counters (applicable for G2 modules only):

Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                                  <snip>
     p6   0                0                0                0
     p7   0                0                0                0
```

Syntax: show ip interface ve <*vid*> **statistics [ethernet** <*port-id*>**]**

Specify the< *port id*> of the virtual interface for which you want to display.

TABLE 46 show ip interface ve statistics command output details

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying detailed IP virtual interface counters

To display the detailed aggregate count of the routed packets and switched packets of a virtual IP interface are configured separate mode, use the following command.

```
NetIron# show ip interface ve 10 statistics detail

Extended Routed Counters (applicable for G2 modules only):

Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                                  <snip>
     p6   0                0                0                0
     p7   0                0                0                0

eth 12/2  0                0                0                0
     p0   0                0                0                0
                              <snip>
     p7   0                0                0                0
```

To display the detailed aggregate count of the routed packets and switched packets of a virtual IP interface are configured combined mode, use the following command.

```
NetIron# show ip interface ve 10 statistics detail

Extended Routed/Switched  Counters (applicable for G2 modules only):

Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
```

```
p1   0                0                0                0
                                <snip>
p6   0                0                0                0
p7   0                0                0                0
```

Syntax: show ip interface ve <*vid*> **statistics [detail]**

Use the <*vid*> *option* to specify the interface name of the virtual IP interface for which you want to display the routed and switched packets aggregate count.

TABLE 47 show ip interface ve output details

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Clearing IP interface counters

When clearing IP interface counters, the counters that are cleared are dependent on the accounting mode configuration. If you have configured accounting mode to count routed and switched packets separately, only the routed counters will be cleared. If you have configured accounting mode to count routed and switched packets together, the combined counter will be cleared.

To clear the routed packet and switched packet counters of a specific IP interface, enter the following command.

```
PowerConnect# clear ip interface ethernet 1/2 statistics
```

Syntax: clear ip interface ethernet <*port-id*> **statistics**

Use the <*port-id*> option to specify the interface name to clear.

Clearing IP virtual interface counters

When clearing IP virtual interface counters, the counters that are cleared are dependent on the accounting mode configuration. If you have configured accounting mode to count routed and switched packets separately, only the routed counters will be cleared. If you have configured accounting mode to count routed and switched packets together, the combined counter will be cleared.

To clear the routed packet and switched packet counters of a specific virtual IP interface, enter the following command

```
PowerConnect# clear ip interface ve 2 statistics
```

Syntax: clear ip interface ve <*vid*> **statistics**

Use the <*vid*> option to specify the virtual interface name to clear.

Transparent VLAN flooding

You can configure your PowerConnect router for transparent VLAN flooding. This feature allows packets to be forwarded without any form of CPU intervention including MAC learning and MAC destination lookups.

NOTE
Because this feature floods all VLAN packets in hardware, it is not expected to work in conjunction with routing functions such as establishing routing protocol neighborship and L3 forwarding even when the VLAN has a VE configured.

This implementation of Transparent VLAN Flooding has the following attributes:

- The ability to always distribute traffic to all members of a VLAN in hardware
- It requires no CPU intervention and consequently can handle line-rate traffic forwarding
- Because this feature does not use any MAC address entries in the CAM it is useful when MAC address entries need to be conserved.
- VLAN members can be tagged or untagged ports including a mix of tagged and untagged ports.
- The maximum number of Transparent VLAN Flooded instances is 4k (default: 8)
- You can mix and match ports with different speeds
- Other Layer 2 capabilities such as spanning tree are unaffected
- Output Layer 3 ACLs may be associated with each port that is part of the VLAN instance being transparently flooded
- You cannot configure a VE interface on a VLAN when Transparent VLAN Flooding is used

This feature is particularly useful in situations where MAC learning is not required for traffic forwarding. Examples of where this feature is useful include:

- A configuration where there are only 2 ports in a VLAN
- Where traffic is looped back to a device through another VLAN for firewall or mirroring purposes
- Where the number of MAC addresses will significantly overwhelm the memory and compute resources of a system.

NOTE
Packets that arrive on an interface with the same destination MAC address as the interface are forwarded in hardware just like packets with other destination addresses.

To enable VLAN transparent forwarding on VLAN 10, use the following command.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# transparent-hw-flooding
```

Syntax: [no] transparent-hw-flooding

- Layer 2 inbound ACLs may be applied to any port in a VLAN that has TVF enabled.
- Layer 2 or Layer 3 outbound ACLs may be applied to any port in a VLAN that has TVF enabled.
- Input Layer 3 ACLs need to be applied to a virutal routing interface on the VLAN and should not be attached to a port directly. Note that the ACL would apply to all ports in the VLAN by default. If this is not desired, a subset of ports in the VLAN may be specified, as in the following configuration.

```
NetIron(config)# interface ve 1
NetIron(config-vif-1)# ip access-group 101 in ethernet 1/1
NetIron(config-vif-1)# ip access-group ve-traffic
```

NOTE
The above example binds the ACL 101 to the virtual routing interface and configures the virtual routing interface to apply the input ACL 101 for switched and routed traffic received on port 1/1.

Displaying VLAN information

After you configure the VLANs, you can view and verify the configuration.

Displaying VLAN information

Enter the **show vlan** command under the vlan-policy configuration.

NOTE
VLAN byte counters are displayed in the output of the **show vlan** command on an MPLS enabled VE interface.

```
NetIron(config-vlan-policy)#show vlan

Configured PORT-VLAN entries: 7
Maximum PORT-VLAN entries: 512
Default PORT-VLAN id: 1

PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level 0, Priority Force 0
Topo HW idx    : 65535    Topo SW idx: 257    Topo next vlan: 0
L2 protocols   : NONE
Untagged Ports : ethe 1/2 to 1/10 ethe 1/12 to 1/20 ethe 3/3
Bytes received : 112329

PORT-VLAN 13, Name [None], Priority Level 0, Priority Force 0
Topo HW idx    : 65535    Topo SW idx: 257    Topo next vlan: 0
L2 protocols   : NONE
Tagged Ports   : ethe 3/2
Bytes received : 112330

PORT-VLAN 23, Name [None], Priority Level 0, Priority Force 0
Topo HW idx    : 65535    Topo SW idx: 257    Topo next vlan: 0
L2 protocols   : NONE
Tagged Ports   : ethe 3/4
Bytes received : 112338
```

Syntax: **show vlan** [<*vlan-id*>] [**|** [**begin** <*expression*> **|** **exclude** <*expression*> **|** **include** <*expression*>]

The output shows the following information.

TABLE 48 Output of show vlan

This field...	Displays...
Configured PORT-VLAN entries	Number of port-based VLANs in the configuration.
Maximum PORT-VLAN entries: 4090	Maximum number of port-based VLANs that you can configure.
Default PORT-VLAN id	ID of the default VLAN.
PORT-VLAN	ID of the port-based VLAN
Name	Name of the port-based VLAN. [None] appears if a name has not been assigned.
Priority Level	Priority level assigned to the port-based VLAN
Priority Force	The priority force is configured on the ingress port with a priority value between 0 and 7. The default is 0. The priority force option allows you to force the priority on the ingress pram, or choose not to enforce the priority.
Topo HW idx	A topology hardware index is a unique hardware ID that is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The range for <*hw-index*> is 0 – 511.
Topo SW idx	The topology group id associated with the VLAN.
Topo next vlan	Next VLAN id in the topology group.
L2 protocols	Layer 2 control protocol configured on the VLAN
Untagged or Tagged Ports	ID of the untagged or tagged ports that are members of the VLAN
Bytes received	Displays the number of received bytes across all ports for a specified VLAN. By default, the vlan-accounting command is turned on and hence the Bytes received field is also displayed by default. However, if VLAN accounting for a VLAN is turned off, then the Bytes received field is not displayed in the output. For more information on enabling VLAN byte counters, refer to "Extended VLAN counters for 8x10G modules" on page 265.

To display information for a specific VLAN, enter a VLAN id as shown in the example below.

```
NetIron(config-vlan-13)#show vlan 13

PORT-VLAN 13, Name [None], Priority Level 0, Priority Force 0
Topo HW idx    : 65535    Topo SW idx: 257    Topo next vlan: 0
L2 protocols   : NONE
Tagged Ports   : ethe 3/2
----------------------------------------------------------
Port  Type      Tag-Mode  Protocol  State
3/2   PHYSICAL  TAGGED    NONE      FORWARDING
Arp Inspection: 0
DHCP Snooping: 0
Multicast Snooping: Disabled
```

Displaying VLAN information for specific ports

To determine which VLANs a port is a member of, enter the following command.

```
NetIron# show vlan e 4/1
Port 4/1 is a member of 2 VLANs
VLANs 1 100
```

Syntax: **show vlan ethernet** <*slot-number*>/<*port-number*> **[| [begin** <*expression*> **| exclude** <*expression*> **| include** <*expression*>**]**

The **ethernet** <*slot-number*>/<*port-number*> parameter specifies a port. The command lists all the VLAN memberships for the port.

The output shows the following information.

TABLE 49 Output of show vlan Ethernet

This field...	Displays...
Port <*slot-number*>/<*port-number*> is a member of # VLANs	The number of VLANs a port is a member of.
VLANs	The IDs of the VLANs that the port is a member of.

Displaying VLAN status and port types

To display detailed information about the state, port types, port modes, of a VLAN, as well as control protocols configured on the VLAN, enter the following command.

```
NetIron# show vlan detail
Untagged Ports  : ethernet 2/1 to 2/20 ethernet 4/4
Tagged Ports    : None
Dual-mode Ports : ethernet 3/1 to 3/20jjj ethernet 4/1 to 4/3
Default VLAN    : 1
Control VLAN    : 4095
VLAN Tag-type   : 0x8100
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
----------------------------------------------------------
Port  Type      Tag-Mode  Protocol  State
2/1   PHYSICAL  UNTAGGED  NONE      DISABLED
2/2   PHYSICAL  UNTAGGED  NONE      DISABLED
2/3   PHYSICAL  UNTAGGED  NONE      DISABLED
2/4   PHYSICAL  UNTAGGED  NONE      DISABLED
2/5   PHYSICAL  UNTAGGED  NONE      DISABLED
.
. (output edited for brevity)
.
4/1   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/2   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/3   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/4   PHYSICAL  UNTAGGED  NONE      DISABLED
PORT-VLAN 100, Name [None], Priority Level0
----------------------------------------------------------
Port  Type      Tag-Mode  Protocol  State
4/1   PHYSICAL  TAGGED    STP       FORWARDING
4/2   PHYSICAL  TAGGED    STP       BLOCKING
```

Syntax: **show vlan detail** <*vlan-id*> **[| [begin** <*expression*> **| exclude** <*expression*> **| include** <*expression*>**]**

Enter the ID of a VLAN if you want information for a specific VLAN.

The output shows the following information.

TABLE 50 Output of show vlan detail

This field...	Displays...
Untagged Ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as untagged ports in all the VLANs on the device.
Tagged Ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as tagged ports in all the VLANs on the device.
Dual-mode ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as dual-mode ports in all the VLANs on the device.
Default VLAN	ID of the default VLAN
Control VLAN	ID of the control VLAN
PORT-VLAN #, Name, Priority Level	Information for each VLAN in the output begins with the VLAN type and its ID, name and priority level. Then ports that are members of the VLAN are listed, with the following information:
Port	Port <slot-number/port-number >
Type	Port type: physical or LAG
Tag-Mode	Tag mode of the port: untagged, tagged, or dual-mode
Protocol	Protocol configured on the VLAN.
State	Current state of the port such as disabled, blocking, forwarding, etc.

Displaying VLAN group information

To display information about VLAN groups, enter the following command.

```
NetIron# show vlan-group 10

Configured VLAN-Group entries : 1
Maximum VLAN-Group entries : 32

VLAN-GROUP 10
Number of VLANs: 4
VLANs: 10 to 13
Tagged ports: ethernet 3/1
```

Syntax: **show vlan-group** [*vlan-group-id*] [**|** [**begin** <*expression*> **|** **exclude** <*expression*> **|** **include** <*expression*>]

The output shows the following information.

TABLE 51 Output of show vlan Ethernet

This field...	Displays...
Configured VLAN-Group entries	Number of VLAN groups that have been configured on the device.
Maximum VLAN-Group entries	Maximum number of VLAN groups that can be configured on the device.
VLAN-Group #	ID of the VLAN group
VLANs	VLANs that belong to the VLAN group.
Tagged ports:	Type and ID of the tagged ports that are members of the VLAN group

Configuring multi-port static MAC address

To configure multi-port static MAC address, enter the following command at the VLAN configuration node level of the CLI. You must configure at least one port.

```
NetIron# vlan 10
static-mac-address 0100.5e42.7f40 multi-ports ethe 1/1 to 1/3
ethe 2/1 to 2/5 priority 5
```

Syntax: [no] static-mac-address <*mac-address*> multi-ports ethernet [<*slot1/port1*>|[<*slot1/port1*> to <*slot1/port#k*>] .. ethernet [<*slot#n/port#n*> to <*slot#n/port#m*>] [priority <*0-7*>]

Limitations

The configuration of multi-port static MAC address has the following limitations:

- A maximum number of 400 multi-port MAC addresses and static MAC addresses can be configured in a system; however, the number of configured entries is limited by the number of multicast FIDs available in the system.
- FIDs with the same port mask are shared among multiple multi-port MAC addresses and multi-port ARP entries to conserve the number of multicast FIDs created in the system.
- Multi-port static MAC address is not supported on POS ports.
- Multi-port static MAC address cannot be configured on VLAN groups.
- You cannot add any of the interface MAC address as multi-port static MAC address.
- Multi-port static MAC address can be configured for either unicast or multicast addresses.
- Unicast MAC addresses configured as multi-port static MAC addresses will not be learned dynamically in the system or allowed to be dynamically moved to a different port.
- If the multi-port static MAC address being added already exists in the dynamic MAC table, then the dynamic MAC will be deleted and replaced with the configured multi-port static MAC.
- Trunk load-balancing is not supported. The multi-port static MAC address traffic will always be forwarded on the active primary port of the trunk port.

Error messages

Error messages are displayed in the following cases:

- You can configure multi-port static MAC addresses only if all the ports in the port list are members of the VLAN.

  ```
  NetIron(config-vlan-100)#static-mac 0001.0001.0003 multi-port eth 2/11
  Error - Multiport mac cannot be configured, Port 2/11 is not member of vlan 100
  ```

- You must provide the complete port mask for deleting the multi-port static MAC address configuration.

  ```
  NetIron(config-vlan-100)#no static-mac 0001.0001.0002 multi-ports eth 2/19 to
  2/20
  Error - the port list does not match with the configured multiport mac port
  list
  ```

- On a trunk port, multi-port static MAC address configuration is allowed only on the primary port of the trunk.

```
NetIron(config-vlan-100)#static-mac-address 0001.0001.0003 multi-ports ethe
2/19 to 2/20 ethe 4/3
Error - Multiport mac cannot be configured with non-primary trunk port  4/3
```

- A port with multi-port static MAC address configuration cannot be a secondary member of the trunk group.

```
NetIron(config-lag-LAG1)#ports eth 4/11
Error: port 4/11 is part of multiport-mac and cannot be added as secondary
port of a trunk
```

- When a LAG primary port is part of a multi-port static MAC address, the LAG cannot be undeployed or deleted. Also, when a non-LAG port is part of a multi-port static MAC address, you cannot deploy a LAG with that port. These configurations will be rejected. The following are the sample error messages for each of these cases:
  - Veto check for deploying LAG.

```
NetIron(config-lag-LAG1)#deploy
Error: LAG LAG1 primary port 4/11 is configured as part of a multi-port mac
entry, cannot deploy the LAG
```

  - Veto check for undeploying LAG.

```
NetIron(config-lag-LAG2)#no deploy
Error: The primary port 4/15 is configured as part of a multi-port mac
entry, cannot undeploy the LAG
```

  - Veto check for deleting LAG.

```
NetIron(config)#no lag LAG2
Error: The primary port 4/15 is configured as part of a multi-port mac
address, cannot remove the LAG
```

- A port belonging to a multi-port static MAC address is not allowed to be removed from a VLAN unless it is removed from all the multi-port static MAC addresses.

```
NetIron(config-vlan-100)#no tag e 2/19
Error - port 2/19 is configured as part of a multi-port mac address, cannot
remove port from vlan
```

- Module configuration deletion (no module) will be rejected if any of the ports in that module are configured as part of any multi-port static MAC addresses.

```
NetIron(config)#no mod 4 ni-mlx-20-port-1g-100fx
Error - module 4 has ports that are member of the multi-port mac addresses,
Cannot remove the module
```

Displaying multi-port static MAC address information

You can display the following information about multi-port static MAC addresses on the device:

- Running configuration
- Changes in the MAC table
- M-port debug information
- LP information

Displaying running configuration

To display the running configuration information of multi-port static MAC addresses, enter the following command.

```
NetIron# show run
vlan 10
tagged ethe 1/1 to 1/20 ethe 2/1 to 2/48
static-mac-address 0100.5e42.7f40 multi-ports ethe 1/1 to 1/3
ethe 2/1 to 2/5 prority 5
```

Displaying changes in the MAC table

You can display the complete MAC table, a specific entry in the MAC table, or a specific MAC entry with M-port details.

To display the complete MAC table, enter the following command.

```
NetIron# show mac
MAC Address     Port      Age      VLAN    Type
0100.5e42.7f40  1/1...    Static   10
Ports  : e 1/1 to 1/3 e 2/1 to 2/5
0000.999d.9996  2/20     Static    10
```

To display a specific MAC address from the MAC table, enter the following command.

```
NetIron# show mac 0100.5e42.7f40
MAC Address     Port      Age      VLAN    Type
0100.5e42.7f40  1/1...   Static   10
Ports: e 1/1 to 1/3 e 2/1 to 2/5
```

To display a specific MAC address with M-port details, enter the following command.

```
NetIron#show mac 0100.5e42.7f40 debug
MAC Address     Port      Age      VLAN    Type
0100.5e42.7f40  1/1…   Static   10
Mport: 30324 FID: 0x00008006 Ports: e 1/1 to 1/3 e 2/1 to 2/5
```

SA and DA learning and aging

Static MAC addresses and multi-port MAC addresses can always be programmed in the CAM of all LP modules with valid VLAN membership. Multi-port static MAC addresses are added or removed from the CAM when a port is added to or deleted from a VLAN. These addresses cannot be dynamically learned or moved to a different port. These addresses will not be removed from the hardware unless the user deletes the multi-port static MAC addresses.

MP switchover and hitless upgrade

The Multi-port static MAC Address feature supports MP switchover and hitless upgrade.

The static MAC table is maintained on both active and standby MPs. The static MAC table is synched to standby MP by CLI configuration commands. The M-port table and FID are also synched from the active MP to the standby MP. After MP switchover, M-port MAC addresses are associated with the FID and, in the case of a missing FID, a new FID will be created and programmed for the multi-port static MAC address.

The static MAC table and FID table will be reprogrammed after LP reload.

Flooding features

User-configured multi-port static MAC addresses will always be programmed on DA CAMs in all PPCR CAMs. So, traffic with this DA MAC address will never be flooded in the VLAN, even when flooding features like transparent VLAN, unknown unicast flooding, multicast flooding, and CPU protection are configured on the system.

Configuring Quality of Service for the NetIron MLX 9

This chapter describes how QoS is implemented and configured in the PowerConnect. The chapter contains the following sections:

- **Ingress Traffic Processing through a Router** – This section describes the QoS operation on Ingress Traffic of a PowerConnect router. Refer to "Ingress Traffic processing through a router" on page 284.
- **Creating an Ingress Decode Policy Map** – This section describes the policy maps used to determine the priority and drop precedence values that will be assigned to the packet within the router. Refer to "Creating an Ingress decode policy map" on page 286.
- **Forcing or Merging Priority of a Packet** – This section describes how once a packet's Ingress priority has been mapped, the values used for processing on the router are determined by either forcing or merging. Refer to "Forcing or merging the priority of a packet" on page 286
- **Forcing or Merging the Drop Precedence of a Packet** – This section describes how once a packet's Ingress drop precedence has been mapped, the values used for processing on the router are determined by either forcing or merging. Refer to "Forcing or merging the drop precedence of a packet" on page 287.
- **Egress Traffic Processing Exiting a Router** – This section describes the QoS operation on Egress Traffic of a PowerConnect router. This process involves marking packets as they leave a router on the Egress port. Refer to "Egress Traffic processing exiting a router" on page 288.
- **Creating an Egress Decode Policy Map** – This section describes the policy maps used to determine the priority and drop precedence values that a packet will carry when it exits the router. Refer to "Creating an egress encode policy map" on page 288.
- **Default QoS Mappings** – This section describes the default mappings for the following PCP Encode, PCP Decode, DSCP Encode, DSCP Decode, EXP Encode and EXP Decode. Refer to "Default QoS mappings" on page 288.
- **Configuring QoS** – This section describes all of the procedures required for both Ingress and Egress QoS Configuration. Refer to "Configuring QoS" on page 294.
- **Displaying QoS Information** – This section describes how to enable and display the following QoS information: QoS Configuration Information and QoS Packet and Byte Statistics. Refer to "Displaying QoS information" on page 321.
- **Configuring Port-level QoS Commands on LAG Ports** – When applying port-level QoS commands to ports in a LAG, the rules can be different according the configuration as described in "Configuring port-level QoS commands on LAG ports" on page 320.
- **Determining Packet Drop Priority using WRED** – Weighted Random Early Detection (WRED) provides a mechanism for determining which packets to drop in a congested network. This section describes how WRED works. Refer to "Weighted Random Early Discard (WRED)" on page 326.
- **Configuring Packet Drop Priority using WRED** – This section describes how to configure Weighted Random Early Detection (WRED). Refer to "Configuring packet drop priority using WRED" on page 328.

- **Scheduling Traffic for Forwarding** – The NetIron supports six different schemes for prioritizing traffic for forwarding in a congested network. This section describes each of these schemes and how to configure them. Refer to "Scheduling traffic for forwarding" on page 334.

Ingress Traffic processing through a router

The QoS operation on Ingress Traffic of a PowerConnect router involves reception and processing of packets based upon priority information contained within the packet. As the packets are processed through the router, there are several opportunities to influence the processing by configuration as described in the steps below.

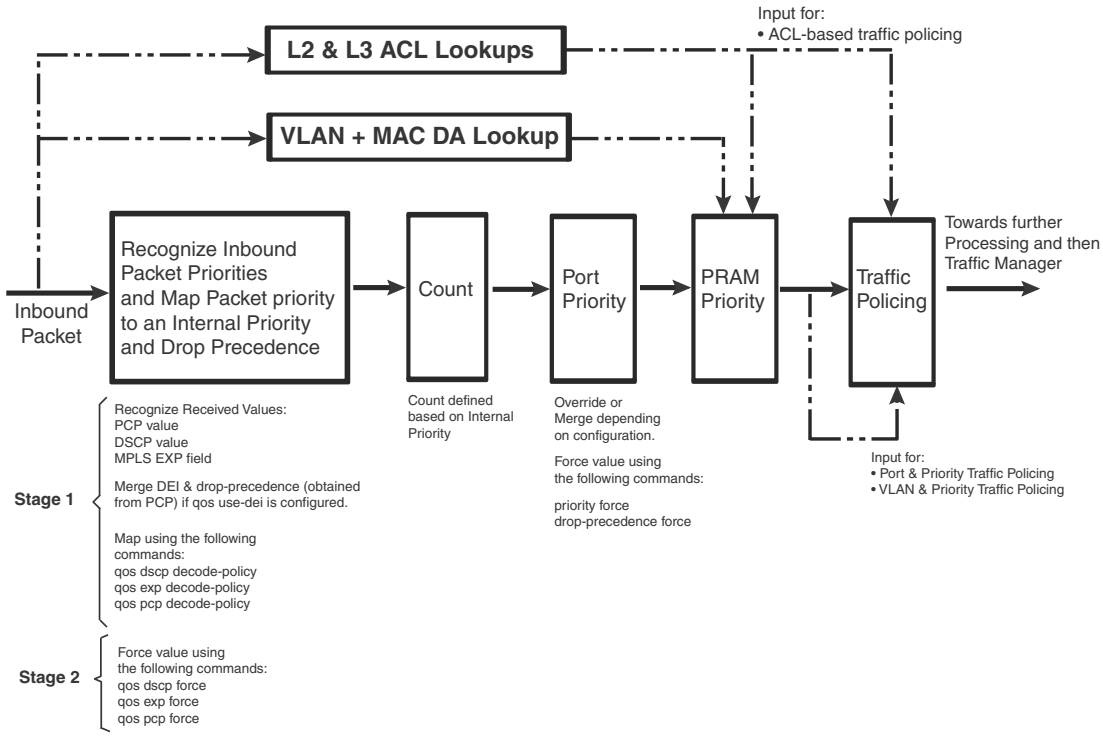
1. Derive priority and drop precedence from the packets PCP (IEEE 802.1p) value. The Priority Code Point (PCP) is a 3-bit field within an IEEE 802.1Q tagged frame that is used to convey the priority of the frame. By using a mapping table, the 3-bit PCP field can be decoded to derive priority and drop precedence information.

NOTE
The PCP field was formerly called IEEE 802.1p.

2. Derive priority and drop precedence from the packets EXP value.
3. Derive priority and drop precedence from the packets DSCP value.
4. Merge or force the priorities described is steps 1 through 3.
5. Merge or force the priority and drop precedence value based on the value configured for the physical port.
6. Merge or force the priority value based on the value configured for the VLAN.
7. Merge or force the priority value based on an ACL look-up. This is used for setting a a specific priority for and L2, L3 or L4 traffic flow.

This process is described in Figure 16.

FIGURE 16 Logic flow of Ingress QoS processing



Recognizing inbound packet priorities and mapping to internal priority

The processes performed in the first block of Figure 16 can be described in two stages as described in the following:

Stage 1

Collect priority and drop precedence information from various portions of the packet header

- If a packet's EtherType matches 8100 or the port's EtherType, derive a priority value and drop precedence by decoding the PCP value.
- If the **qos use-dei** command is configured, the bit between the VLAN ID and PCP in the VLAN tag will be interpreted as a drop precedence and priority value.
- For MPLS packets, derive a priority value and drop precedence by decoding the EXP bits.
- For IPv4 or v6 packets, derive a priority value and drop precedence by decoding the DSCP bits.
- The derived values for PCP, EXP and DSCP are mapped to either a default map or a configured Ingress Decode Policy Map.
- To assist the router in the decoding process described in "stage 1" decode-map tables are defined.

Stage 2
Determine if a priority value should be forced or merged

- If a packet's EtherType matches 8100 or the port's EtherType, derive a priority value and drop precedence by decoding the PCP value
- If the **qos pcp force** command is configured on the port, the priority and drop precedence values are set to the value read from the PCP bits.
- If the **qos exp force** command is configured on the port, the priority and drop precedence values are set to the value read from the MPLS EXP bits.
- If the **qos dscp force** command is configured on the port, the priority and drop precedence values are set to the value read from the DSCP bits.
- If none of the qos force commands are configured, the priority and drop precedence values are set for IPv4 or v6 packets and MPLS packets as described in the following:

  For IPv4 or v6 Packets: Priority and drop precedence values obtained as a merge of the decoded PCP and decoded DSCP values.

  For MPLS Packets: Priority and drop precedence values obtained as a merge of the decoded PCP and decoded EXP values.

Creating an Ingress decode policy map

Once a packet's Ingress priority has been recognized for the PCP, DSCP and EXP values, those values are matched against a policy map to determine the priority and drop precedence values that will be assigned to the packet within the router. The maps used can be either:

- Default policy maps described in "Default QoS mappings" on page 288, or
- User-configured policy maps that are defined as described:

  dscp decode-map <*decode-map-name*> – This command allows you to map a recognized DSCP value to a value that you define.

  pcp decode-map <*decode-map-name*> – This command allows you to map a recognized PCP value to a value that you define.

  exp decode-map <*decode-map-name*> – This command allows you to map a recognized MPLS EXP value to a value that you define.

Forcing or merging the priority of a packet

Once a packet's Ingress priority has been mapped, the values that will be used for processing on the router are determined by either forcing or merging.

There are a variety of commands to "force" the priority of a packet based on the following criteria:

- Forced to a priority configured for a specific Ingress port. The **priority force** command is configured at the interface where you want it to be applied.
- Forced to a priority configured for a specific VLAN. The **priority force** command is configured at the VLAN where you want is to be applied.
- Forced to a priority that is obtained from the DSCP priority bits. The **qos- dscp force** command is configured at the interface where you want it to be applied.

- Forced to a priority that is obtained from the EXP priority bits. The **qos- exp force** command is configured at the interface where you want it to be applied.
- Forced to a priority that is obtained from the PCP priority bits. The **qos- pcp force** command is configured at the interface where you want it to be applied.
- Forced to a priority that is based on an ACL match. The **priority-force** keyword can be used within an ACL to apply a priority to specified traffic.

If multiple commands containing the **priority force** keyword are specified, the command with the highest precedence will take effect as determined by the following order.

1. ACL match (if the **qos-tos mark cos** command is configured, it has the same #1 priority precedence as ACL match). Refer to "Specifying the trust level and enabling marking" on page 316 for details.
2. VLAN priority
3. Physical port priority value
4. DSCP value in an incoming IPv4 or v6 packet
5. EXP value in an incoming MPLS packet
6. PCP value in a tagged frame or PCP field or .1ad DE

Details of how to configure the force commands are provided in "Configuring a force priority" on page 305. For information about configuring a force priority with ACLs refer to the following:

- "Filtering and priority manipulation based on 802.1p priority" on page 817 (for Layer-2 ACLs)
- "QoS for IPv6 traffic" on page 1746

Forcing or merging the drop precedence of a packet

Once a packet's Ingress drop precedence has been mapped, the values that will be used for processing on the router are determined by either forcing or merging.

There are a variety of commands to "force" the drop precedence of a packet based on the following criteria:

- Forced to a drop precedence configured for a specific Ingress port. The **drop-precedence force** command is configured at the interface where you want it to be applied.
- Forced to a drop precedence that is obtained from the DSCP priority bits. The **qos dscp force** command is configured at the interface where you want it to be applied.
- Forced to a drop precedence that is obtained from the EXP priority bits. The **qos exp force** command is configured at the interface where you want it to be applied.
- Forced to a drop precedence that is obtained from the PCP priority bits. The **qos pcp force** command is configured at the interface where you want it to be applied.
- Forced to a drop precedence that is based on an ACL match. The **drop-precedence-force** keyword can be used within an ACL to apply a priority to specified traffic.

If multiple commands containing the **force** keyword are specified, the command with the highest precedence will take effect as determined by the following order.

1. ACL match
2. Physical port's drop precedence value

3. DSCP value in an incoming IPv4v6 packet
4. EXP value in an incoming MPLS packet
5. PCP value in a tagged frame or PCP field or .1ad DE

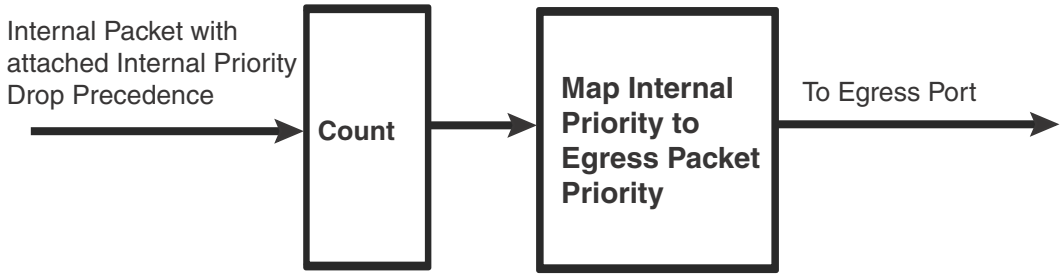
Details of how to configure the force commands are provided in "Configuring a force priority" on page 305. For information about configuring a force priority with ACLs refer to the following:

- "Filtering and priority manipulation based on 802.1p priority" on page 817 (for Layer-2 ACLs)
- "QoS for IPv6 traffic" on page 1746

Egress Traffic processing exiting a router

The QoS operation on Egress Traffic of a PowerConnect router involves marking packets as they leave a router on the egress port. As the packets are prepared to exit the router you can set the PCP, DSCP, and EXP values in the packet headers. This process is described in Figure 17.

FIGURE 17 Logic flow of Egress QoS processing



Creating an egress encode policy map

The QoS value that a packet carries in its header when it exits a PowerConnect router on an egress interface is determined by a specified mapping. Unless configured, this value once determined is placed in an internal queue by using one of the default maps described in "Default QoS mappings" on page 288. Alternately, the following commands can be used to define an alternate mapping:

- **pcp encode-map** <*encode-map-name*> – This command allows you to map the internal priority and drop precedence values of a packet into the PCP code point.
- **dscp encode-map** <*encode-map-name*> – This command allows you to map the internal priority and drop precedence values of a packet into the DSCP code point.
- **exp encode-map** <*encode-map-name*> – This command allows you to map the internal priority and drop precedence values of a packet into the EXP code point.

Default QoS mappings

If a user defined map is not created or applied to Ingress or Egress traffic, the PowerConnect router uses a default map to assign PCP, DSCP and EXP priority and drop precedence values. The following tables describe the default QoS mapping values:

- PCP Encode Table

- PCP Decode Table
- DSCP Encode Table
- DSCP Decode Table
- EXP Encode Table
- EXP Decode Table

Table 52 lists the default PCP Encode mappings.

NOTE
The encode commands for QOS mapping such as **qos pcp encode** <on | off>, **qos dscp encode** <on | off> and **qos exp encode** <on | off> are available only at the physical port level. There are no direct commands at global level to enable or disable QOS encode mapping.

TABLE 52 PCP encode table

Priority & Drop Eligibility (DE)		7	7DE	6	6DE	5	5DE	4	4DE	3	3DE	2	2DE	1	1DE	0	0DE
	8P0D (default)	7	7	6	6	5	5	4	4	3	3	2	2	1	1	0	0
PCP	7P1D	7	7	6	6	5	4	5	4	3	3	2	2	1	1	0	0
	6P2D	7	7	6	6	5	4	5	4	3	2	3	2	1	1	0	0
	5P3D	7	7	6	6	5	4	5	4	3	2	3	2	1	0	1	0

Table 53 lists the default PCP Decode mappings.

TABLE 53 PCP decode table

PCP		7	6	5	4	3	2	1	0
	8P0D (default)	7	6	5	4	3	2	1	0
PCP	7P1D	7	6	4	4DE	3	2	1	0
	6P2D	7	6	4	4DE	2	2DE	1	0
	5P3D	7	6	4	4DE	2	2DE	0	0DE

Table 54 lists the default DSCP Encode mappings.

TABLE 54 Default DSCP encode table

Priority decimal (binary)	Drop-precedence decimal (binary)	DSCP decimal (binary)	Priority decimal (binary)	Drop-precedence decimal (binary)	DSCP decimal (binary)
0 (000)	0 (00)	0 (000000)	4 (100)	0 (00)	32 (100000)
0 (000)	1 (01)	2 (000010)	4 (100)	1 (01)	34 (100010)
0 (000)	2 (10)	4 (000100)	4 (100)	2 (10)	36 (100100)
0 (000)	3 (11)	6 (000110)	4 (100)	3 (11)	38 (100110)
1 (001)	0 (00)	8 (001000)	5 (101)	0 (00)	40 (101000)
1 (001)	1 (01)	10 (001010)	5 (101)	1 (01)	42 (101010)
1 (001)	2 (10)	12 (001100)	5 (101)	2 (10)	44 (101100)

TABLE 54 Default DSCP encode table (Continued)

Priority decimal (binary)	Drop-precedence decimal (binary)	DSCP decimal (binary)	Priority decimal (binary)	Drop-precedence decimal (binary)	DSCP decimal (binary)
1 (001)	3 (11)	14 (001110)	5 (101)	3 (11)	46 (101110)
2 (010)	0 (00)	16 (010000)	6 (110)	0 (00)	48 (110000)
2 (010)	1 (01)	18 (010010)	6 (110)	1 (01)	50 (110010)
2 (010)	2 (10)	20 (010100)	6 (110)	2 (10)	52 (110100)
2 (010)	3 (11)	22 (010110)	6 (110)	3 (11)	54 (110110)
3 (011)	0 (00)	24 (011000)	7 (111)	0 (00)	56 (111000)
3 (011)	1 (01)	26 (011010)	7 (111)	1 (01)	58 (111010)
3 (011)	2 (10)	28 (011100)	7 (111)	2 (10)	60 (111100)
3 (011)	3 (11)	30 (011110)	7 (111)	3 (11)	62 (111110)

Table 55 and Table 56 list the default DSCP Decode mappings.

TABLE 55 Default DSCP decode table

DSCP decimal (binary)	Priority decimal (binary)	Drop-precedence decimal (binary)	DSCP decimal (binary)	Priority decimal (binary)	Drop-precedence decimal (binary)
0 (000000)	0 (000)	0 (00)	16 (010000)	2 (010)	0 (00)
1 (000001)	0 (000)	0 (00)	17(010001)	2 (010)	0 (00)
2 (000010)	0 (000)	1 (01)	18 (010010)	2 (010)	1 (01)
3 (000011)	0 (000)	1 (01)	19 (010011)	2 (010)	1 (01)
4 (000100)	0 (000)	2 (10)	20 (010100)	2 (010)	2 (10)
5 (000101)	0 (000)	2 (10)	21(010101	2 (010)	2 (10)
6 (000110)	0 (000)	3 (11)	22 (010110)	2 (010)	3 (11)
7 (000111)	0 (000)	3 (11)	23 (010111)	2 (010)	3 (11)
8 (001000)	1 (001)	0 (00)	24 (011000)	3 (011)	0 (00)
9 (001001)	1 (001)	0 (00)	25 (011001)	3 (011)	0 (00)
10 (001010)	1 (001)	1 (01)	26 (011010)	3 (011)	1 (01)
11 (001011)	1 (001)	1 (01)	27 (011011)	3 (011)	1 (01)
12 (001100)	1 (001)	2 (10)	28 (011100)	3 (011)	2 (10)
13 (001101	1 (001)	2 (10)	29 (011101)	3 (011)	2 (10)
14 (001110)	1 (001)	3 (11)	30 (011110)	3 (011)	3 (11)
15 (001111)	1 (001)	3 (11)	31 (011111)	3 (011)	3 (11)

TABLE 56 Default DSCP decode table (cont.)

DSCP decimal (binary)	Priority decimal (binary)	Drop-precedence decimal (binary)	DSCP decimal (binary)	Priority decimal (binary)	Drop-precedence decimal (binary
32 (100000)	4 (100)	0 (00)	48 (110000)	6 (110)	0 (00)
33 (100001)	4 (100)	0 (00)	49 (110001)	6 (110)	0 (00)
34 (100010)	4 (100)	1 (01)	50 (110010)	6 (110)	1 (01)
35 (100011)	4 (100)	1 (01)	51(110011)	6 (110)	1 (01)
36 (100100)	4 (100)	2 (10)	52(110100)	6 (110)	2 (10)
37 (100101)	4 (100)	2 (10)	53(110101)	6 (110)	2 (10)
38 (100110)	4 (100)	3 (11)	54 (110110)	6 (110)	3 (11)
38 (100111)	4 (100)	3 (11)	55 (110111)	6 (110)	3 (11)
40 (101000)	5 (101)	0 (00)	56 (111000)	7 (111)	0 (00)
41 (101001)	5 (101)	0 (00)	57 (111001)	7 (111)	0 (00)
42 (101010)	5 (101)	1 (01)	58 (111010)	7 (111)	1 (01)
43 (101011)	5 (101)	1 (01)	58 (111011)	7 (111)	1 (01)
44 (101100)	5 (101)	2 (10)	60 (111100)	7 (111)	2 (10)
45 (101101)	5 (101)	2 (10)	61 (111101)	7 (111)	2 (10)
46 (101110)	5 (101)	3 (11)	62 (111110)	7 (111)	3 (11)
47(101111)	5 (101)	3 (11)	63 (111111)	7 (111)	3 (11)

Table 54 lists the default EXP Encode mappings. Please note that software forwarded VPLS packets do not use the EXP encode table.

TABLE 57 Default EXP encode table

Priority decimal (binary)	Drop-precedence decimal (binary)	EXP value	Priority decimal (binary)	Drop-precedence decimal (binary)	DSCP decimal (binary)
0 (000)	0 (00)	0	4 (100)	0 (00)	4
0 (000)	1 (01)	0	4 (100)	1 (01)	4
0 (000)	2 (10)	0	4 (100)	2 (10)	4
0 (000)	3 (11)	0	4 (100)	3 (11)	4
1 (001)	0 (00)	1	5 (101)	0 (00)	5
1 (001)	1 (01)	1	5 (101)	1 (01)	5
1 (001)	2 (10)	1	5 (101)	2 (10)	5
1 (001)	3 (11)	1	5 (101)	3 (11)	5
2 (010)	0 (00)	2	6 (110)	0 (00)	6
2 (010)	1 (01)	2	6 (110)	1 (01)	6
2 (010)	2 (10)	2	6 (110)	2 (10)	6
2 (010)	3 (11)	2	6 (110)	3 (11)	6
3 (011)	0 (00)	3	7 (111)	0 (00)	7

TABLE 57 Default EXP encode table

Priority decimal (binary)	Drop-precedence decimal (binary)	EXP value	Priority decimal (binary)	Drop-precedence decimal (binary)	DSCP decimal (binary)
3 (011)	1 (01)	3	7 (111)	1 (01)	7
3 (011)	2 (10)	3	7 (111)	2 (10)	7
3 (011)	3 (11)	3	7 (111)	3 (11)	7

Table 54 lists the default EXP Encode mappings

TABLE 58 Default EXP encode table

Priority decimal (binary)	Drop-precedence decimal (binary)	EXP value	Priority decimal (binary)	Drop-precedence decimal (binary)	EXP value
0 (000)	0 (00)	0	4 (100)	0 (00)	4
0 (000)	1 (01)	0	4 (100)	1 (01)	4
0 (000)	2 (10)	0	4 (100)	2 (10)	4
0 (000)	3 (11)	0	4 (100)	3 (11)	4
1 (001)	0 (00)	1	5 (101)	0 (00)	5
1 (001)	1 (01)	1	5 (101)	1 (01)	5
1 (001)	2 (10)	1	5 (101)	2 (10)	5
1 (001)	3 (11)	1	5 (101)	3 (11)	5
2 (010)	0 (00)	2	6 (110)	0 (00)	6
2 (010)	1 (01)	2	6 (110)	1 (01)	6
2 (010)	2 (10)	2	6 (110)	2 (10)	6
2 (010)	3 (11)	2	6 (110)	3 (11)	6
3 (011)	0 (00)	3	7 (111)	0 (00)	7
3 (011)	1 (01)	3	7 (111)	1 (01)	7
3 (011)	2 (10)	3	7 (111)	2 (10)	7
3 (011)	3 (11)	3	7 (111)	3 (11)	7

Table 54 lists the default EXP Decode mappings

TABLE 59 Default EXP decode table

EXP value	Priority decimal (binary)	Drop-precedence decimal (binary)
7	7 (111)	0
6	6 (110)	0
5	5 (101)	0
4	4 (100)	0
3	3 (011)	0

TABLE 59 Default EXP decode table

EXP value	Priority decimal (binary)	Drop-precedence decimal (binary)
2	2 (010)	0
1	2 (001)	0
0	0 (000)	0

Protocol Packet Prioritization

Certain control packets are handled with certain priorities by default and hence those priorities cannot be lowered with any of the QoS configuration commands or the priority force command. The list of these control packets are listed below.

Table 60 on page 293 lists the protocol packets that are internally and automatically prioritized for IPv4, L2, and IPv6.

TABLE 60 Default prioritized protocol table

Protocol Packets
IPv4/L2
ARP
STP/RSTP/BPDU
MRP
VSRP
LACP
GARP
UDLD
IGMP
OSPF / OSPF over GRE
BGP / BGP over GRE
RIP
IS-IS
ES-IS
VRRP
VRRPE
PIM / PIM over GRE
DVMRP
MSDP / MSDP over GRE
RSVP
LDP basic

TABLE 60 Default prioritized protocol table

Protocol Packets
LDP extended
BOOTP/DHCP
IPv4 Router Alert
ISIS over GRE or GRE
Keep Alive Packets
BFD (Bidirectional Forwarding Detection)
IPv6
OSPF / OSPF in 6to4
BGP / BGP in 6to4
RIPNG
MLD
ND6 / ND6 in 6to4
VRRP
VRRPE PIM / PIM in 6to4 BFD (Bidirectional Forwarding Detection)
PIM / PIM in 6to4
BFD (Bidirectional Forwarding Detection)

Configuring QoS

The QoS configuration process involves separate procedures for Ingress and Egress QoS Processing as described in the following major sections.

Configuring Ingress QoS procedures

The following procedures are required to configure a PowerConnect router for Ingress QoS processing:

- **Creating Ingress Decode Policy Maps** – If you want the priority and drop precedence values used within the router to be mapped to a specified value, you must create a Decode Priority map as described in "Configuring Ingress decode policy maps" on page 295.
- **Binding Ingress Decode Policy Maps** – If you want to apply an Ingress Policy Map other than the default, you must bind the Ingress Policy Map either globally or to a specified interface as described in "Binding Ingress decode policy maps" on page 301.

- **Configuring a Force priority** – Where there are multiple QoS values that can be used to determine the QoS level used on the router, the default policy is to determine the value used by performing a merge as described in "Stage 2 Determine if a priority value should be forced or merged" on page 286. Otherwise, you can specify a value that you want used from either the port or VLAN configured value or the DSCP, EXP or PCP values in the packets as described in "Configuring a force priority" on page 305.

Configuring Egress QoS procedures

The following procedures are required to configure a PowerConnect router for Egress QoS processing:

- **Creating Egress Encode Policy Maps** – If you want the priority and drop precedence values of packets leaving the router to be marked with a specified value, you must create an Encode Priority map as described in "Configuring Egress encode policy maps" on page 307.
- **Binding Egress Encode Policy Maps** – If you want to apply an Egress Policy Map other than the default, you must bind the Egress Policy Map either globally or to a specified interface as described in "Binding an Egress encode EXP policy map" on page 311.

Configuring QoS procedures applicable to Ingress and Egress

The following procedures are required to configure procedures applicable to both Ingress and Egress QoS Processing on a PowerConnect router:

- **Enabling a Port to Use the DEI bit** – You can configure the router to use the DEI bit when computing the drop precedence value for an incoming packet or encoding the DEI bit for transmitted frame as described in "Enabling a port to use the DEI bit for Ingress and Egress processing" on page 316.
- **Specifying the Trust Level and Enabling Marking** – Refer to "Specifying the trust level and enabling marking" on page 316
- **Support for Super Aggregate VLANS** – If you want to use the enhanced QoS feature with Super Aggregate VLANs, refer to "Configuring support for super aggregate VLANs" on page 320
- **Support for QoS Configurations on LAG Ports** – If you are configuring the enhanced QoS feature on ports within a LAG, refer to "Configuring port-level QoS commands on LAG ports" on page 320

Configuring Ingress decode policy maps

Ingress Decode Policy Maps are created globally and are applied later either globally for all ports on a router or locally to specific port. To create an Ingress Decode Policy Map, you must first enter the QoS mapping configuration level of the command interface using the **qos-mapping** command, as shown in the following.

```
NetIron(config)# qos-mapping
```

Configuration of each of the decode and encode mappings is described in the following sections:

- Configuring Ingress Decode DSCP Policy Maps
- Configuring Ingress Decode PCP Policy Maps

- Configuring Ingress Decode EXP Policy Maps

Configuring Ingress decode DSCP policy maps

The following procedures are used when configuring an Ingress Decode DSCP Policy Map:

- Naming an Ingress Decode DSCP Policy Map
- Configuring an Ingress Decode DSCP Policy Map

Naming an Ingress decode DSCP policy map

Once you are in the QoS configuration level, can define the name of a Ingress Decode DSCP Policy Map using the **pcp decode-map** command, as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# dscp decode-map Customer1
```

Syntax: [no] dscp decode-map <*map-name*>

The **no** option is used to delete a currently configured Ingress Decode DSCP Policy Map. If the Ingress Decode DSCP Policy Map is currently in use the **no** command will be rejected and an error message will be displayed.

The <*map-name*> variable specifies the name of the Ingress Decode DSCP Policy Map that you are defining. It can be up to 64 characters in length. You can specify the same <*map-name*> for different types of policy maps. For example, you can use the same name for an Ingress Decode DSCP Policy Map and a Ingress Decode EXP Policy Map.

NOTE
The name "default-map" cannot be used because it is reserved for standard mappings as described in "Default QoS mappings" on page 288.

Configuring an Ingress decode DSCP policy map

Once you have named an Ingress Decode DSCP Policy Map using the **dscp decode-map** command, you can set the values of the named Ingress Decode DSCP Policy Map. Setting the values in an Ingress Decode DSCP Policy Map involves specifying the value of the DSCP bits of an incoming packet and setting them to correspond to a value of 0 to 7 of the router's internal priority. Optionally, you can set a drop precedence value of 0 to 3 in addition to the internal priority value.

To set the values of an Ingress Decode DSCP Policy Map, first specify name of the policy map and then populate the values in the policy map using the **dscp-value** command as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# dscp decode-map Customer1
NetIron(config-qos-mapping-dscp-decode)# dscp-value 32 to priority 5
drop-precedence 2
```

Syntax: [no] dscp-value <*dscp-value*> [<*dscp-value*>] **to priority** <*priority-value*>
[drop-precedence <*dp-value*>]

The <*dscp-value*> variable specifies the value of the DSCP bits within the packet header of the incoming packets. You can optionally specify multiple <*dscp-value*> variables if you want to specify more than one value to map to the same internal priority and drop precedence. Where DSCP values within a policy map are unspecified, the default mapping will be used.

The **priority** keyword together with the <*priority-value*> variable specifies the internal priority that the packets with the previously specified <*dscp-value*> value will be mapped to. The <*priority-value*> variable can be a value between 0 and 7. Please note, when generating the configuration file, a configured priority value that is the same as the value in the default priority map will not be shown.

The **drop-precedence** keyword is an optional parameter that allows you to specify a <*dp-number*> variable that represents the drop precedence value that you want to assign to incoming packets with the previously specified <*dp-value*> value. This value is specified in addition to a **priority** <*priority-value*> value. The <*dp-number*> variable can be a value between 0 and 3. The default value is the value described in the default DSCP table. Please note, when generating the configuration file, a value for drop precedence will only be shown for non-default values.

When using the **[no]** option to negate a previously configured value of this command, observe the considerations described below.

1. You can negate both the **priority** and **drop-precedence** values (returning them to their default values) by using the **[no]** option with the original command only up to the **priority** value.

   For example: the following command has been used to set the map to assign an internal priority of "4" and a drop precedence of "2" to Ingress packets that have a DSCP value of "40".

   ```
   NetIron(config-qos-mapping-dscp-decode)# dscp-value 40 to priority 4
   drop-precedence 2
   ```

   To set the priority and and **drop-precedence** values back to the default values, use the **[no]** option with the previous command up to where the **priority** value is configured, as shown in the following.

   ```
   NetIron(config-qos-mapping-dscp-decode)# no dscp-value 40 to priority 4
   ```

   After this command is executed, the **priority** and **drop-precedence** values for **dscp-value 40** will be returned to their default values as described in the default map tables that are defined in "Default QoS mappings" on page 288.

2. You can negate the **drop-precedence** value (returning it to its default value) without changing the currently configured **priority** value. This is done by using the **[no]** option with the original command that includes both the **priority** and **drop-precedence** values.

   For example: the following command has been used to set the priority map to assign an internal priority of "5" and a drop precedence of "1" to Ingress packets that have a DSCP value of "60".

   ```
   NetIron(config-qos-mapping-dscp-decode)# dscp-value 60 to priority 5
   drop-precedence 1
   ```

   To set the **drop-precedence** value back to the default value, use the **[no]** option with the previous command, as shown in the following.

   ```
   NetIron(config-qos-mapping-dscp-decode)# no dscp-value 60 to priority 5
   drop-precedence 1
   ```

   After this command is executed, the **priority** value will remain at 5 and the **drop-precedence** value will be returned to the default **drop-precedence** value for **dscp-value 60**, as described in the default map tables that are defined in "Default QoS mappings" on page 288.

Configuring Ingress decode PCP policy maps

The following procedures are used when configuring an Ingress Decode PCP Policy Map:

- Naming an Ingress Decode PCP Policy Map
- Configuring an Ingress Decode PCP Policy Map

Naming an Ingress decode PCP policy map

Once you are in the QoS configuration level, can define the name of an Ingress Decode PCP Policy Map using the **dscp decode-map** command, as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# pcp decode-map Customer1
```

Syntax: [no] pcp decode-map <*map-name*>

The **no** option is used to delete a currently configured Ingress decode PCP policy map. If the policy map is currently in use, the **no** command will be rejected and an error message will be displayed.

The <*map-name*> variable specifies the name of the Ingress decode PCP policy map that you are defining. It can be up to 64 characters in length. You can specify the same map name for different types of maps. For example, you can use the same name for an Ingress decode PCP policy map and an Ingress decode DSCP policy map.

NOTE
The name "default-map" cannot be used because it is reserved for standard mappings as described in "Default QoS mappings" on page 288.

Configuring an Ingress decode PCP policy map

Once you have named an Ingress PCP Decode Policy Map using the **pcp decode-map** command, you can set the values of the named policy map. Setting the values in a policy map involves specifying the value of the PCP bits of an incoming packet and setting them to correspond to a value of 0 to 7 of the router's internal priority. Optionally, you can set a drop precedence value of 0 to 3 in addition to the internal priority value.

To set the values of an Ingress Decode PCP Policy Map, first specify name of the policy map and then populate the values in the Ingress Decode PCP Policy Map using the **pcp-value** command as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# pcp decode-map Customer1
NetIron(config-qos-mapping-pcp-decode)# pcp-value 7 to priority 3 drop-precedence
2
```

Syntax: [no] pcp-value <*pcp-value*> [<*pcp-value*>] **to priority** <*priority-value*> **[drop-precedence** <*dp-value*>]

The <*pcp-value*> variable specifies the value of the PCP bits within the packet header of the incoming packets. You can optionally specify multiple <*pcp-value*> variables if you want to specify more than one value to map to the same internal priority and drop precedence. Where PCP values within a policy map are unspecified, the default mapping will be used.

The **priority** keyword together with the <*priority-value*> variable specifies the internal priority that the packets with the previously specified <*pcp-value*> value will be mapped to. The <*priority-value*> variable can be a value between 0 and 7. Please note, when generating the configuration file a configured priority value that is the same as the value in the default map will not be shown.

The **drop-precedence** keyword is an optional parameter that allows you to specify a <*dp-number*> variable that represents the drop precedence value that you want to assign to incoming packets with the previously specified <*dp-value*> value. This value is specified in addition to a **priority** <*priority-value*> value. The <*dp-number*> variable can be a value between 0 and 3. The default value is 0. Please note, when generating the configuration file a value for drop precedence will only be shown for non-zero values.

When using the **[no]** option to negate a previously configured value of this command, observe the considerations described below.

1. You can negate both the **priority** and **drop-precedence** values (returning them to their default values) by using the **[no]** option with the original command only up to the **priority** value.

   For example: the following command has been used to set the map to assign an internal priority of "3" and a drop precedence of "2" to Ingress packets that have a PCP value of "7".

   ```
   NetIron(config-qos-mapping-pcp-decode)# pcp-value 7 to priority 3
   drop-precedence 2
   ```

   To set the **priority** and and **drop-precedence** values back to the default values, use the **[no]** option with the previous command up to where the **priority** value is configured, as shown in the following.

   ```
   NetIron(config-qos-mapping-pcp-decode)# no pcp-value 7 to priority 3
   ```

   After this command is executed, the **priority** and **drop-precedence** values for **pcp-value 7** will be returned to their default values as described in the default map tables that are defined in "Default QoS mappings" on page 288.

2. You can negate the **drop-precedence** value (returning it to its default value) without changing the currently configured **priority** value. This is done by using the **[no]** option with the original command that includes both the **priority** and **drop-precedence** values.

   For example: the following command has been used to set the priority map to assign an internal priority of "4" and a drop precedence of "2" to Ingress packets that have a PCP value of "6".

   ```
   NetIron(config-qos-mapping-pcp-decode)# pcp-value 6 to priority 4
   drop-precedence 2
   ```

   To set the **drop-precedence** value back to the default value, use the **[no]** option with the previous command, as shown in the following.

   ```
   NetIron(config-qos-mapping-pcp-decode)# no pcp-value 6 to priority 4
   drop-precedence 2
   ```

   After this command is executed, the **priority** value will remain at 4 and the **drop-precedence** value will be returned to the default **drop-precedence** value for **pcp-value 6**, as described in the default map tables that are defined in "Default QoS mappings" on page 288.

Configuring Ingress decode EXP Ppolicy maps

The following procedures are used when configuring an Ingress Decode EXP Policy Map:

- Naming an Ingress Decode EXP Policy Map
- Configuring an Ingress Decode EXP Policy Map

Naming an Ingress decode EXP policy mMap

Once you are in the QoS configuration level, can define the name of a Ingress Decode EXP Policy Map using the **exp decode-map** command, as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# exp decode-map Customer1
```

Syntax: [no] exp decode-map <*map-name*>

The **no** option is used to delete a currently configured Ingress Decode EXP Policy Map. If the Ingress Decode EXP Policy Map is currently in use, the **no** command will be rejected and an error message will be displayed.

The <*map-name*> variable specifies the name of the Ingress Decode EXP Policy Map that you are defining. It can be up to 64 characters in length. You can specify the same Ingress Decode EXP Policy Map for different types of policy maps. For example, you can use the same name for an Ingress Decode DSCP Policy Map and an Ingress Decode EXP Policy Map.

NOTE
The name "default-map" cannot be used because it is reserved for standard mappings as described in "Default QoS mappings" on page 288.

Configuring an Ingress decode EXP policy map

Once you have named an Ingress Decode EXP Policy Map using the **exp decode-map** command, you can set the values of the named policy map. Setting the values in a policy map involves specifying the value of the EXP bits of an incoming packet and setting them to correspond to a value of 0 to 7 of the router's internal priority value. Optionally, you can set a drop precedence value of 0 to 3 in addition to the internal priority value.

To set the values of an Ingress Decode EXP Policy Map, first specify name of the policy map and then populate the values in the policy map using the **exp-value** command as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# exp decode-map Customer1
NetIron(config-qos-mapping-exp-decode)# exp-value 7 to priority 5 drop-precedence
2
```

Syntax: [no] exp-value <*exp-value*> **[<***exp-value***>] to priority <***priority-value***> [drop-precedence <***dp-value***>]**

The <*exp-value*> variable specifies the value of the EXP bits within the packet header of the incoming packets. You can optionally specify multiple <*exp-value*> variables if you want to specify more than one value to map to the same internal priority and drop precedence values. Where EXP values within a policy map are unspecified, the default mapping will be used.

The **priority** keyword together with the <*priority-value*> variable specifies the internal priority value that the packets with the previously specified <*exp-value*> value will be mapped to. The <*priority-value*> variable can be a value between 0 and 7. Please note, when generating the configuration file a configured priority value that is the same as the value in the default priority map will not be shown.

The **drop-precedence** keyword is an optional parameter that allows you to specify a <*dp-number*> variable that represents the drop precedence value that you want to assign to incoming packets with the previously specified <*dp-value*> value. This value is specified in addition to a **priority** <*priority-value*> value. The <*dp-number*> variable can be a value between 0 and 3. The default value is the value described in the default EXP table. Please note, when generating the configuration file a value for drop precedence will only be shown for non-default values.

When using the **[no]** option to negate a previously configured value of this command, observe the considerations described below.

1. You can negate both the **priority** and **drop-precedence** values (returning them to their default values) by using the **[no]** option with the original command only up to the **priority** value.

   For example: the following command has been used to set the map to assign an internal priority of "5" and a drop precedence of "2" to Ingress packets that have an EXP value of "7".

   ```
   NetIron(config-qos-mapping-exp-decode)# exp-value 7 to priority 5
   drop-precedence 2
   ```

   To set the **priority** and and **drop-precedence** values back to the default values, use the **[no]** option with the previous command up to where the **priority** value is configured, as shown in the following.

   ```
   NetIron(config-qos-mapping-exp-decode)# no exp-value 7 to priority 5
   ```

   After this command is executed, the **priority** and **drop-precedence** values for **exp-value 7** will be returned to their default values as described in the default map tables that are defined in "Default QoS mappings" on page 288.

2. You can negate the **drop-precedence** value (returning it to its default value) without changing the currently configured **priority** value. This is done by using the **[no]** option with the original command that includes both the **priority** and **drop-precedence** values.

   For example: the following command has been used to set the priority map to assign an internal priority of "5" and a drop precedence of "2" to Ingress packets that have a EXP value of "7".

   ```
   NetIron(config-qos-mapping-exp-decode)# exp-value 7 to priority 5
   drop-precedence 2
   ```

   To set the **drop-precedence** value back to the default value, use the **[no]** option with the previous command, as shown in the following.

   ```
   NetIron(config-qos-mapping-exp-decode)# no exp-value 7 to priority 5
   drop-precedence 2
   ```

   After this command is executed, the **priority** value will remain at 5 and the **drop-precedence** value will be returned to the default **drop-precedence** value for **exp-value 7**, as described in the default map tables that are defined in "Default QoS mappings" on page 288.

Binding Ingress decode policy maps

You can bind an Ingress decode policy map globally or per-port using either the default policy map, an all zero policy map, or a user defined policy map. Additionally, for PCP, you can bind the following pre-defined policy maps: 7P1D, 6P2D, and 5P3D. The following procedures describe how to bind Ingress decode policy maps:

- Binding Ingress Decode DSCP Policy Maps
- Binding Ingress Decode PCP Policy Maps
- Binding Ingress Decode EXP Policy Maps

Binding Ingress decode DSCP policy maps

The following procedures describe how to configure the binding of Ingress Decode DSCP Policy Maps:

- Globally Binding an Ingress Decode DSCP Policy Map
- Binding an Ingress Decode DSCP Policy Map to a Port

Globally Binding an Ingress decode DSCP policy map

You can bind an Ingress Decode DSCP Policy Map globally for a PowerConnect router using the **qos dscp decode-policy** command as shown in the following.

```
NetIron(config)# qos dscp decode-policy Customer1
```

Syntax: [no] qos dscp decode-policy <*decode-map-name*> **| default-map | all-zero-map**

The <*decode-map-name*> variable is the name assigned to the Ingress Decode DSCP Policy Map that you want applied globally on the router. If you try to apply an <*decode-map-name*> value that has not been defined, the configuration will be rejected. If the <*decode-map-name*> value that has been defined but the policy has not been configured, the configuration will be accepted and the **default-map** will be applied.

The **default-map** option assigns the default Ingress Decode DSCP Policy Map globally on the router. Since the default Ingress Decode DSCP Policy Map is the default setting, this option is only required when the router has been previously set to a different Ingress Decode DSCP Policy Map.

The **all-zero-map** option assigns a Ingress Decode DSCP Policy Map where all DSCP values are mapped to priority 0 and drop precedence 0. This is useful if you do not want to process any DSCP information in the incoming packet.

Binding an Ingress decode DSCP policy map to a port

You can bind an Ingress Decode DSCP Policy Map to a specified port on a PowerConnect router using the **qos dscp decode-policy** command within an interface configuration, as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos dscp decode-policy Customer1
```

Syntax: [no] qos dscp decode-policy <*decode-map-name*> **| default-map | all-zero-map**

The <*decode-map-name*> variable is the name assigned to the Ingress Decode DSCP Policy Map that you want applied to the port whose configuration this is under.

The **default-map** option assigns the default Ingress Decode DSCP Policy Map to the port whose configuration this is under. Since the default Ingress Decode DSCP Policy Map is the global default setting, this option is only required when the router's global map has been set to a Ingress Decode DSCP Policy Map other than the default.

The **all-zero-map** option assigns an Ingress Decode DSCP Policy Map where all DSCP values are mapped to priority 0 and drop precedence 0. This is useful if you do not want to process any DSCP information in the incoming packet

Binding an Ingress decode PCP policy map

The following procedures describe how to configure the binding of an Ingress Decode PCP Policy Map:

- Globally Binding an Ingress Decode PCP Policy Map
- Binding an Ingress Decode PCP Policy Map to a Port

Globally binding an Ingress decode PCP policy map

You can bind an Ingress Decode PCP Policy Map globally for a PowerConnect router using the **qos pcp decode-policy** command as shown in the following.

```
NetIron(config)# qos pcp decode-policy Customer1
```

Syntax: [no] qos pcp decode-policy *<decode-map-name>* **| default-map | all-zero-map | 7P1D |6P2D | 5P3D**

The *<decode-map-name>* variable is the name assigned to the Ingress Decode PCP Policy Map that you want applied globally on the router. If you try to apply an *<decode-map-name>* value that has not been defined, the configuration will be rejected. If the *<decode-map-name>* value that has been defined but the policy has not been configured, the configuration will be accepted and the **default-map** will be applied.

The **default-map** option assigns the default Ingress Decode PCP Policy Map globally on the router. The default policy map for PCP is the 8P0D decode map. Since the default Ingress Decode PCP Policy Map is the default setting, this option is only required when the router has been previously set to a different Ingress Decode PCP Policy Map.

The **all-zero-map** option assigns an Ingress Decode PCP Policy Map where all PCP values are mapped to priority 0 and drop precedence 0. This is useful if you do not want to process any PCP information in the incoming packet.

The **7P1D** option assigns the 7P1D Ingress Decode PCP Policy Map globally on the router.

The **6P2D** option assigns the 6P2D Ingress Decode PCP Policy Map globally on the router.

The **5P3D** option assigns the 5P3D Ingress Decode PCP Policy Map globally on the router.

NOTE
7P1D, 6P2D, and **5P3D** are as defined in the IEEE 802.1ad specification and described in Table 53.

Binding an Ingress decode PCP policy map to a port

You can bind an Ingress Decode PCP Policy Map to a specified port on a PowerConnect router using the **qos pcp decode-policy** command as shown in the following.

```
NetIron(config)# qos pcp decode-policy Customer1
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos pcp decode-policy Customer1
```

Syntax: [no] qos pcp decode-policy <*decode-map-name***> | default-map | all-zero-map | 7P1D |6P2D | 5P3D**

The *<decode-map-name>* variable is the name assigned to the Ingress Decode PCP Policy Map that you want applied to the port whose configuration this is under.

The **default-map** option assigns the default Ingress Decode PCP Policy Map to the port whose configuration this is under. Since the default Ingress Decode PCP Policy Map is the default setting, this option is only required when the router's global map has been set to an Ingress Decode PCP Policy Map other than the default.

The **all-zero-map** option assigns an Ingress Decode PCP Policy Map where all PCP values are mapped to priority 0 and drop precedence 0. This is useful if you do not want to process any PCP information in the incoming packet.

The **7P1D** option assigns the 7P1D Ingress Decode PCP Policy Map to the port whose configuration this is under.

The **6P2D** option assigns the 6P2D Ingress Decode PCP Policy Map to the port whose configuration this is under.

The **5P3D** option assigns the 5P3D Ingress Decode PCP Policy Map to the port whose configuration this is under.

NOTE
7P1D, 6P2D, and **5P3D** are as defined in the IEEE 802.1ad specification and described in Table 53.

Binding an Ingress decode EXP policy map

The following procedures describe how to configure the binding of an Ingress Decode EXP Policy Map:

- Globally Binding an Ingress Decode EXP Policy Map
- Binding an Ingress Decode EXP Policy Map to a Port

Globally binding an Ingress decode EXP policy map

You can bind an Ingress Decode EXP Policy Map globally for a PowerConnect router using the **qos exp decode-policy** command as shown in the following.

```
NetIron(config)# qos exp decode-policy Customer1
```

Syntax: [no] qos exp decode-policy <*decode-map-name*> | default-map | all-zero-map

The <*decode-map-name*> variable is the name assigned to the Ingress Decode EXP Policy Map that you want applied globally on the router. If you try to apply a <*decode-map-name*> value that has not been defined, the configuration will be rejected. If the <*decode-map-name*> value that has been defined but the policy has not been configured, the configuration will be accepted and the **default-map** will be applied.

The **default-map** option assigns the default Ingress Decode EXP Policy Map globally on the router. Since the default Ingress Decode EXP Policy Map is the default setting, this option is only required when the router has been previously set to a different Ingress Decode EXP Policy Map.

The **all-zero-map** option assigns an Ingress Decode EXP Policy Map where all EXP values are mapped to priority 0 and drop precedence 0.

Binding an Ingress decode EXP policy map to a port

You can bind an Ingress Decode EXP Policy Map to a specified port on a PowerConnect router using the **qos exp decode-policy** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos exp decode-policy Customer1
```

Syntax: [no] qos exp decode-policy <*decode-map-name*> | default-map | all-zero-map

The <*decode-map-name*> variable is the name assigned to the Ingress Decode EXP Policy Map that you want applied to the port whose configuration this is under,

The **default-map** option assigns the default Ingress Decode EXP Policy Map to the port whose configuration this is under. Since the default Ingress Decode EXP Policy Map is the default setting, this option is only required when the router's global map has been set to an Ingress Decode EXP Policy Map other than the default.

The **all-zero-map** option assigns an Ingress Decode EXP Policy Map where all EXP values are mapped to priority 0 and drop precedence 0. This is useful if you do not want to process any EXP information in the incoming packet

Configuring a force priority

In situations where there are conflicting priority values for packets on an Ingress port, that conflict can be resolved by performing a priority merge or by using a **force** command to direct the router to use a particular value above other values. A **force** command can be configured for each of the following:

- Force to the values configured on a port
- Force to the value configured for a VLAN
- Force to the value in the DSCP bits
- Force to the value in the EXP bits
- Force to the value in the PCP bits
- Force to a value specified within an ACL

Configuring a force priority for a port

You can configure an Ingress port with a priority to apply to packets that arrive on it using the **priority** command.

To configure an Ingress port with a priority, use the **priority** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)priority 6
```

Syntax: [no] priority < *priority-value*>

The <*priority-value*> variable is a value between 0 and 7. The default value is 0.

Once a port has been configured with a priority using the **priority** command, you can then configure the port (using the **priority force** command) to force the configured priority when determining the priority relative to other priority values of incoming packets.

To configure an Ingress port to force the port-configured priority, use the **priority force** command as shown in the following:

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)priority force
```

Syntax: [no] priority force

Configuring a force drop precedence for a port

You can configure an Ingress port with a drop precedence to apply to packets that arrive on it using the **drop-precedence** command.

To configure an Ingress port with a drop precedence, use the **drop-precedence** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)drop-precedence 3
```

Syntax: [no] drop-precedence <*dp-value*>

The <*dp-value*> variable is a value between 0 and 3.

Once a port has been configured with a drop precedence using the **drop-precedence** command, you can then configure the port (using the **drop-precedence force** command) to force the configured drop precedence when determining the priority relative to other priority values of incoming packets.

To configure an Ingress port to force the port-configured drop precedence, use the **drop-precedence force** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)drop-precedence force
```

Syntax: [no] drop-precedence force

Configuring a force priority for a VLAN

By default, VLANs have priority 3. To change a port-based VLAN's QoS priority, use the following method. The priority applies to outbound traffic on ports in the VLAN.

To change the QoS priority of port-based VLAN 20 on a Chassis device to priority queue 7, enter the following commands.

```
NetIron(config)# vlan 20
NetIron(config-vlan-20)# priority 7
```

Syntax: [no] priority <*num*>

The <*num*> parameter can be from 3 – 7 and specifies one of the eight QoS queues.

Once a VLAN has been configured with a priority using the **priority** command, you can then configure the VLAN (using the **priority force** command) to force the configured priority when determining the priority relative to other priority values of incoming packets.

To configure an Ingress port to force the VLAN-configured priority, use the **priority force** command as shown in the following.

```
NetIron(config)# vlan 20
NetIron(config-vlan-20) priority force
```

Syntax: [no] priority force

Configuring force priority to the DSCP value

You can configure an Ingress port (using the **qos dscp force** command) to force the configured DSCP value when determining the priority relative to other priority values of incoming packets.

To configure an Ingress port to force the DSCP value, use the **qos dscp force** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos dscp force
```

Syntax: [no] qos dscp force

Configuring force priority to the EXP value

You can configure an Ingress port (using the **qos exp force** command) to force the configured EXP value when determining the priority relative to other priority values of incoming packets.

To configure an Ingress port to force the EXP value, use the **qos exp force** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos exp force
```

Syntax: qos exp force

Configuring force priority to the PCP value

You can configure an Ingress port (using the **qos pcp force** command) to force the configured PCP value when determining the priority relative to other priority values of incoming packets.

To configure an Ingress port to force the PCP value, use the **qos pcp force** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos pcp force
```

Syntax: qos pcp force

This command is not supported on POS interfaces.

Configuring force priority to a value specified by an ACL

You can use the **priority-force** keyword within an ACL to apply a priority to specified traffic as described in "Filtering and priority manipulation based on 802.1p priority" on page 778.

Configuring Egress encode policy maps

Egress Encode Policy Maps are created globally and are applied later either globally for all ports on a router or locally to specific port. To create an Egress Encode Policy Map, you must first enter the QoS mapping configuration level of the command interface using the **qos-mapping** command, as shown in the following.

```
NetIron(config)# qos-mapping
```

Configuration of each of the Egress Encode Policy Maps is described in the following sections:

- Configuring Egress Encode DSCP Policy Maps
- Configuring Egress Encode PCP Policy Maps
- Configuring Egress Encode EXP Policy Maps

Configuring Egress encode DSCP policy maps

The following procedures are used when configuring an Egress Encode DSCP Policy Map:

- Naming an Egress Encode DSCP Policy Map
- Configuring an Egress Encode DSCP Policy Map

Naming an Egress encode DSCP policy map

Once you are in the QoS configuration level, can define the name of an Egress Encode DSCP Policy Map using the **dscp encode-map** command, as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# dscp encode-map Customer1
```

Syntax: [no] dscp encode-map <*map-name*>

The **no** option is used to delete a currently configured Egress Encode DSCP Policy Map. If the policy map is currently in use, the **no** command will be rejected and an error message will be displayed.

The <*map-name*> variable specifies the name of the Egress Encode DSCP Policy Map that you are defining. It can be up to 64 characters in length. You can specify the same policy map name for different types of maps. For example, you can use the same policy pam name for an Egress Encode DSCP Policy Map and an Egress Encode EXP Policy map.

NOTE
The name "default-map" cannot be used because it is reserved for standard mappings as described in "Default QoS mappings" on page 288.

Configuring an Egress encode DSCP policy map

Once you have named an Egress Encode DSCP Policy Map using the **dscp encode-map** command, you can set the values of the named encode policy map. Setting the values in an Egress Encode DSCP Policy Map involves specifying a DSCP value to be marked in outgoing packets for a specified priority value (0 - 7) and optionally a drop precedence value (0 - 3).

To set the values of an Egress Encode DSCP Policy Map, first specify name of the policy map and then populate the values in the Egress Encode DSCP Policy Map using the **priority** command as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# dscp encode-map Customer1
NetIron(config-qos-mapping-dscp-encode)# priority 7 drop-precedence 2 to
dscp-value 3
```

Syntax: [no] **priority** <*priority-value*> [**drop-precedence** <*dp-value*>] [<*dp-value*>] **to dscp-value** <*dscp-value*>

The **priority** keyword together with the <*priority-value*> variable specifies the internal priority value that egress packets will be marked from. The <*priority-value*> variable can be a value between 0 and 7. For unspecified priority values, the default mapping values are used.

The **drop-precedence** keyword is an optional parameter that allows you to specify a <*dp-number*> variable that represents the drop precedence value that you specify in addition to a **priority** <*priority-value*> value. The <*dp-number*> variable can be a value between 0 and 3. Multiple <*dp-number*> variables can be configured in a single command. The default value is "any" which means that drop priorities 0 - 3 are assigned. If a drop precedence value is specified only for a subset of values, the entries with unspecified values will be initialized as specified in the default mapping.

The <*dscp-value*> variable specifies the value that will be marked onto the DSCP bits within the packet header of the outgoing packets. This applies to packets that match the **priority** and **drop precedence** values specified in this command.

The **[no]** option allows you to negate a previously configured value and return to the default mapping for the specified **priority** and **drop precedence** values.

Configuring an Egress encode PCP policy map

The following procedures are used when configuring an Egress Encode PCP Policy Maps:

- Naming an Egress Encode PCP Policy Map

- Configuring an Egress Encode PCP Policy Map

Naming an Egress encode PCP policy map

Once you are in the QoS configuration level, can define the name of an Egress Encode PCP Policy Map using the **pcp encode-map** command, as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# pcp encode-map Customer1
```

Syntax: [no] pcp encode-map <*map-name*>

The **no** option is used to delete a currently configured Egress Encode PCP Policy Map. If the policy map is currently in use, the **no** command will be rejected and an error message will be displayed.

The <*map-name*> variable specifies the name of the Egress Encode PCP Policy Map that you are defining. It can be up to 64 characters in length. You can specify the same policy map name for different types of policy maps. For example, you can use the same name for an Egress Encode PCP Policy Map and an Egress Encode EXP Policy Map.

NOTE
The name "default-map" cannot be used because it is reserved for standard mappings as described in "Default QoS mappings" on page 288.

Configuring an Egress encode PCP policy map

Once you have named an Egress Encode PCP Policy Map using the **pcp encode-map** command, you can set the values of the named policy map. Setting the values in an Egress Encode PCP Policy Map involves specifying a PCP value to be marked in outgoing packets for a specified internal priority value (0 - 7) and optionally a drop precedence value (0 - 3).

To set the values of an Egress Encode PCP Policy Map, first specify name of the policy map and then populate the values in the policy map using the **priority** command as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# pcp encode-map Customer1
NetIron(config-qos-mapping-pcp-encode)# priority 7 drop-precedence 2 to pcp-value
3
```

Syntax: [no] priority <*priority-value*> **[drop-precedence** <*dp-value*>**] [**<*dp-value*>**] to pcp-value** <*pcp-value*>

The **priority** keyword together with the <*priority-value*> variable specifies the priority value that the egress packets will be marked with. The <*priority-value*> variable can be a value between 0 and 7. For unspecified priority values, the default mapping values are used.

The **drop-precedence** keyword is an optional parameter that allows you to specify a <*dp-number*> variable that represents the drop precedence value that you specify in addition to a **priority** <*priority-value*> value. The <*dp-number*> variable can be a value between 0 and 3. Multiple <*dp-number*> variables can be configured in a single command. The default value is "any" which means that drop priorities 0 - 3 are assigned. If a drop precedence value is specified only for a subset of values, the entries with unspecified values will be initialized as specified in the default mapping.

The <*pcp-value*> variable specifies the value that will be marked onto the PCP bits within the packet header of the outgoing packets. This applies to packets that match the **priority** and **drop precedence** values specified in this command.

The **[no]** option allows you to negate a previously configured value and return to the default mapping for the specified **priority** and **drop precedence** values.

Configuring an Egress Eencode EXP policy map

The following procedures are used when configuring an Egress Encode EXP Policy Map:

- Naming an Egress Encode EXP Policy Map
- Configuring an Egress Encode EXP Policy Map

Naming an Egress encode EXP policy map

Once you are in the QoS configuration level, can define the name of an Egress Encode EXP Policy Map using the **exp encode-map** command, as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# exp encode-map Customer1
```

Syntax: [no] exp encode-map <*map-name*>

The <*map-name*> variable specifies the name of the Egress Encode EXP Policy Map that you are defining. It can be up to 64 characters in length. You can specify the same policy map name for different types of policy maps. For example, you can use the same name for an Egress Encode EXP Policy and an Egress Encode DSCP Policy Map.

The **no** option is used to delete a currently configure Egress Encode EXP Policy Map. If the policy map is currently in use, the **no** command will be rejected and an error message will be displayed.

NOTE
The name "default-map" cannot be used because it is reserved for standard mappings as described in "Default QoS mappings" on page 288.

Configuring an Egress encode EXP policy map

Once you have named an Egress Encode EXP Policy Map using the **exp encode-map** command, you can set the values of the named encode policy map. Setting the values in an Egress Encode EXP Policy Map involves specifying an EXP value to be marked in outgoing packets for a specified priority value (0 - 7) and optionally a drop precedence value (0 - 3).

To set the values of an Egress Encode EXP Policy Map, first specify name of the policy map and then populate the values in the policy map using the **priority** command as shown in the following.

```
NetIron(config)# qos-mapping
NetIron(config-qos-mapping)# exp encode-map Customer1
NetIron(config-qos-mapping-exp-encode)# priority 7 drop-precedence 2 to exp-value
3
```

Syntax: [no] priority <*priority-value*> **[drop-precedence** <*dp-value*>**] [**<*dp-value*>**] to exp-value** <*exp-value*>

The **priority** keyword together with the <*priority-value*> variable specifies the internal forwarding value of the egress packets. The <*priority-value*> variable can be a value between 0 and 7. For unspecified priority values, the default mapping values are used.

The **drop-precedence** keyword is an optional parameter that allows you to specify a <*dp-number*> variable that represents the drop precedence value that you specify in addition to a **priority** <*priority-value*> value. The <*dp-number*> variable can be a value between 0 and 3. Multiple <*dp-number*> variables can be configured in a single command. The default value is "any" which means that drop priorities 0 - 3 are assigned. If a drop precedence value is specified only for a subset of values, the entries with unspecified values will be initialized as specified in the default mapping.

The <*exp-value*> variable specifies the value that will be marked onto the EXP bits within the packet header of the outgoing packets. This applies to packets that match the **priority** and **drop precedence** values specified in this command.

The **[no]** option allows you to negate a previously configured value and return to the default mapping for the specified **priority** and **drop precedence** values.

Binding an Egress encode EXP policy map

You can bind an Egress Encode Policy map globally or per-port using either the default policy map, an all zero policy map, or a user defined policy map. Additionally, for PCP, you can bind the following pre-defined policy maps: 7P1D, 6P2D, and 5P3D. The following procedures describe how to bind Egress Encode Policy Maps:

- Binding an Egress Encode DSCP Policy Map
- Binding an Egress Encode PCP Policy Map
- Binding an Engress Encode EXP Policy Map

Binding an Egress encode DSCP policy map

The following procedures describe how to configure the binding of an Egress Encode DSCP Policy Map:

- Globally Binding an Egress Encode DSCP Policy Map
- Binding an Egress Encode DSCP Policy Map to a Port

Globally binding an Egress encode DSCP policy map

You can bind an Egress Encode DSCP Policy Map globally for a PowerConnect router using the **qos dscp decode-policy** command as shown in the following.

```
NetIron(config)# qos dscp encode-policy Customer1
```

Syntax: [no] qos dscp encode-policy <*encode-map-name*> | default-map | all-zero-map

The <*encode-map-name*> variable is the name assigned to the Egress Encode DSCP Policy Map that you want applied globally on the router. If you try to apply a <*encode-map-name*> value that has not been defined, the configuration will be rejected. If the <*encode-map-name*> value that has been defined but the policy has not been configured, the configuration will be accepted and the **default-map** will be applied.

The **default-map** option assigns the default Egress Encode DSCP Policy Map globally on the router. Since the default Egress Encode DSCP Policy Map is the default setting, this option is only required when the router has been previously set to a different Egress Encode DSCP Policy Map. When configured globally, the **qos dscp encode-policy default-map** command will not be displayed within the configuration even if it is explicitly configured.

The **all-zero-map** option assigns an Egress Encode DSCP Policy Map where all 32 combinations of priority and drop precedence are mapped to 0.

The **no** option allows you to withdraw a previously configured encode policy. If the **qos dscp encode-policy** command is not configured, then the **no qos pcp encode-policy** command will generate an error message.

The **no** option allows you to withdraw a previously configured encode policy. If the **qos dscp encode-policy default-map** command is not configured, then the **no qos dscp encode-policy default-map** command will still be allowed because the **qos dscp encode-policy default-map** is the default configuration.

Binding an Egress encode DSCP policy map to a port

You can bind an Egress Encode DSCP Policy Map to a specified port on a PowerConnect router using the **qos dscp encode-policy** command within an interface configuration, as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos dscp encode-policy Customer1
NetIron(config-if-e10000-10/1)qos dscp encode-policy on
```

Syntax: [no] qos dscp encode-policy <*encode-map-name*> **| default-map | all-zero-map**

NOTE
The **qos dscp encode-policy on** command is shown in this example because unlike PCP or EXP, the DSCP encode policy is off by default.

The <*encode-map-name*> variable is the name assigned to the Egress Encode DSCP Policy Map that you want applied to the port whose configuration this is under.

The **default-map** option assigns the default Egress Encode DSCP Policy Map to the port whose configuration this is under. Since the default Egress Encode DSCP Policy Map is the global default setting, this option is only required when the router's global map has been set to an Egress Encode DSCP Policy Map other than the default. The **qos dscp encode-policy** command will not be displayed within the configuration unless it is explicitly configured.

The **all-zero-map** option assigns an Egress Encode DSCP Policy Map where all 32 combinations of priority and drop precedence are mapped to 0.

The **no** option allows you to withdraw a previously configured encode policy. If the **qos pcp encode-policy** command is not configured, then the **no qos pcp encode-policy** command will generate an error message.

The **no** option allows you to withdraw a previously configured Egress Encode DSCP Policy Map. If the **qos dscp encode-policy default-map** command is not configured, then the **no qos dscp encode-policy default-map** command will generate an error message because the **qos dscp encode-policy default-map** command was never configured on the port.

Enabling and disabling an Egress Encode DSCP Policy Map on a port

To enable or disable an Egress Encode DSCP Policy Map on a port, use the **qos dscp encode-policy** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos dscp encode-policy on
```

Syntax: [no] qos dscp encode-policy on | off

The **on** option enables DSCP encode on the port. The **qos dscp encode-policy on** command will not be displayed within the configuration unless it is explicitly configured.

The **off** option disables DSCP encode on the port. This is the default setting.

Binding Egress encode PCP policy map

The following procedures describe how to configure the binding of an Egress Encode PCP Policy Map:

- Globally Binding an Egress Encode PCP Policy Map Policy
- Binding an Egress Encode PCP Policy Map to a Port

Globally binding an Egress Encode PCP Policy Map

You can bind an Egress Encode PCP Policy Map globally for a PowerConnect router using the **qos pcp encode-policy** command as shown in the following.

```
NetIron(config)# qos pcp encode-policy Customer1
```

Syntax: [no] qos pcp encode-policy <*encode-map-name*> **| default-map | all-zero-map | 7P1D |6P2D | 5P3D**

The <*encode-map-name*> variable is the name assigned to the Egress Encode PCP Policy Map that you want applied globally on the router. If you try to apply a <*encode-map-name*> value that has not been defined, the configuration will be rejected. If the <*encode-map-name*> value has been defined but the policy has not been configured, the configuration will be accepted and the **default-map** will be applied.

The **default-map** option assigns the default Egress Encode PCP Policy Map globally on the router. Since the default Egress Encode PCP Policy Map is the default setting, this option is only required when the router has been previously set to a different Egress Encode PCP Policy Map. When configured globally, the **qos pcp encode-policy default-map** command will not be displayed within the configuration even if it is explicitly configured.

The **all-zero-map** option assigns an Egress Encode PCP Policy Map where all 32 combinations of priority and drop precedence are mapped to 0.

The **7P1D** option assigns the 7P1D Egress Encode PCP Policy Map globally on the router.

The **6P2D** option assigns the 6P2D Egress Encode PCP Policy Map globally on the router.

The **5P3D** option assigns the 5P3D Egress Encode PCP Policy Map globally on the router.

NOTE
7P1D, 6P2D, and **5P3D** are as defined in the IEEE 802.1ad specification and described in Table 52.

The **no** option allows you to withdraw a previously configured encode policy. If the **qos pcp encode-policy default-map** command is not configured, then the **no qos pcp encode-policy default-map** command will still be allowed because the **qos pcp encode-policy default-map** is the default configuration.

Binding an Egress encode PCP policy map to a port

You can bind an Egress Encode PCP Policy Map to a specified port on a PowerConnect router using the **qos pcp encode-policy** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos pcp encode-policy Customer1
```

Syntax: [no] qos pcp encode-policy <*encode-map-name*> **| default-map | all-zero-map | 7P1D |6P2D | 5P3D**

The <*encode-map-name*> variable is the name assigned to the Egress Encode PCP Policy Map that you want applied to the port whose configuration this is under.

The **default-map** option assigns the default Egress Encode PCP Policy Map to the port whose configuration this is under. Since the default Egress Encode PCP Policy Map is the default setting, this option is only required when the router's global map has been set to an Egress Encode PCP Policy Map other than the default. The **qos pcp encode-policy default-map** command will not be displayed within the configuration unless it is explicitly configured.

The **all-zero-map** option assigns an Egress Encode PCP Policy Map where all 32 combinations of priority and drop precedence are mapped to 0.

The **7P1D** option assigns the 7P1D Egress Encode PCP Policy Map to the port whose configuration this is under.

The **6P2D** option assigns the 6P2D Egress Encode PCP Policy Map to the port whose configuration this is under.

The **5P3D** option assigns the 5P3D Egress Encode PCP Policy Map to the port whose configuration this is under.

NOTE
7P1D, 6P2D, and **5P3D** are as defined in the IEEE 802.1ad specification and described in Table 52.

The **no** option allows you to withdraw a previously configured Egress Encode PCP Policy Map. If the **qos pcp encode-policy** command is not configured, then the **no qos pcp encode-policy** command will generate an error message.

The **no** option allows you to withdraw a previously configured Egress Encode PCP Policy Map. If the **qos pcp encode-policy default-map** command is not configured, then the **no qos pcp encode-policy default-map** command will generate an error message because the **qos pcp encode-policy default-map** command was never configured on the port.

Enabling and disabling an Egress Encode PCP Policy Map on a port

To enable or disable an Egress Encode DSCP Policy Map on a port, use the **qos pcp encode-policy** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos pcp encode-policy on
```

Syntax: qos pcp encode-policy on | off

The **on** option enables PCP encode-policy on the port. This is the default setting. The **qos pcp encode-policy on** command will not be displayed within the configuration unless it is explicitly configured.

The **off** option disables PCP encode-policy on the port.

This command is not supported on POS interfaces.

Binding Egress encode EXP policy maps

The following procedures describe how to configure the binding of an Egress Encode EXP Policy Map:

- Globally Binding an Egress Encode EXP Policy Map
- Binding an Egress Encode EXP Policy Map to a Port

Globally binding an Egress Encode EXP Policy Map

You can bind an Egress Encode EXP Policy Map globally for a PowerConnect router using the **qos exp encode-policy** command as shown in the following.

```
NetIron(config)# qos exp encode-policy Customer1
```

Syntax: [no] qos exp encode-policy *<encode-map-name>* **| default-map | all-zero-map**

The *<encode-map-name>* variable is the name assigned to the Egress Encode EXP Policy Map that you want applied globally on the router. If you try to apply an *<encode-map-name>* value that has not been defined, the configuration will be rejected. If the *<encode-map-name>* value that has been defined but the policy has not been configured, the configuration will be accepted and the **default-map** will be applied.

The **default-map** option assigns the default Egress Encode EXP Policy Map globally on the router. Since the default Egress Encode EXP Policy Map is the default setting, this option is only required when the router has been previously set to a different Egress Encode EXP Policy Map. When configured globally, the **qos exp encode-policy default-map** command will not be displayed within the configuration even if it is explicitly configured.

The **all-zero-map** option assigns an Egress Encode EXP Policy Map where all 32 combinations of priority and drop precedence are mapped to 0.

The **no** option allows you to withdraw a previously configured Egress Encode EXP Policy Map. If the **qos exp encode-policy** command is not configured, then the **no qos exp encode-policy** command will generate an error message.

The **no** option allows you to withdraw a previously configured Egress Encode EXP Policy Map. If the **qos exp encode-policy default-map** command is not configured, the **no qos exp encode-policy default-map** command will still be allowed because **qos exp encode-policy default-map** is the default configuration.

Binding an Egress Encode EXP Policy Map to a port

You can bind an Egress Encode EXP Policy Map to a specified port on a PowerConnect router using the **qos exp encode-policy** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos exp encode-policy Customer1
```

Syntax: [no] qos exp encode-policy *<encode-map-name>* **| default-map | all-zero-map**

The *<encode-map-name>* variable is the name assigned to the Egress Encode EXP Policy Map that you want applied to the port whose configuration this is under,

The **default-map** option assigns the default Egress Encode EXP Policy Map to the port whose configuration this is under. Since the default Egress Encode EXP Policy Map is the default setting, this option is only required when the router's global policy map has been set to an Egress Encode EXP Policy Map other than the default. The **qos exp encode-policy default-map** command will not be displayed within the configuration unless it is explicitly configured.

The **all-zero-map** option assigns an Egress Encode EXP Policy Map where all 32 combinations of priority and drop precedence are mapped to 0.

The **no** option allows you to withdraw a previously configured encode policy. If the **qos exp encode-policy default-map** command is not configured, then the **no qos exp encode-policy default-map** command will generate an error message because the **qos exp encode-policy default-map** command was never configured on the port.

Enabling and disabling an Egress Encode EXP Policy Map on a port

To enable or disable an Egress Encode EXP Policy Map on a port, use the **qos exp encode-policy** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos exp encode-policy on
```

Syntax: [no] qos exp encode-policy on | off

The **on** option enables EXP encode on the port. This is the default setting. The **qos exp encode-policy on** command will not be displayed within the configuration unless it is explicitly configured.

The **off** option disables EXP encode on the port.

Enabling a port to use the DEI bit for Ingress and Egress processing

In the IEEE 802.1ad specification, two types of tag are defined:

- Customer VLAN tag (C-TAG)
- Service VLAN tag (S-TAG)

The semantics and structure of the S-TAG is identical to that of the C-TAG, with the exception that bit 5 in octet 1, the Drop Eligible Indicator (DEI) bit, dis used to indicate if the packet is drop eligible. This allows all 3 bits in the PCP ID to be used for indicating priority of the packet with the drop precedence indicated by the DEI bit. The IEEE 802.1ad requires that if this capability is provided, it must be independently manageable for each port.

On the PowerConnect router the **qos use-dei** command can be configured at the port level to allow a drop-precedence value for incoming packet to be computed based on the DEI bit. Additionally, if this command is configured, then a drop-eligible parameter will be encoded in the DEI bit of transmitted frames. If the internal drop precedence of the packet is 2 or 3, the DEI will be transmitted as 1; otherwise it will be transmitted as 0.

This command is configured as described in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos use-dei
```

Syntax: [no] qos use-dei

NOTE
This command applies for both Ingress and Egress processing.

Specifying the trust level and enabling marking

The commands are as follows:

- qos-tos trust
- qos-tos mark

These commands operate on the QoS values within the packets as they arrive on the router. The **qos-tos trust** command specifes which value among the following to use to classify the packet for marking: **cos, ip-prec,** and **dscp**. The **qos-tos mark** command specifies a CoS or DSCP value to mark on outgoing packets as specified by the mappings described in "Packet mapping commands" on page 318.

NOTES: You cannot use these commands and other L4 features such as:

- IPv4 ACLs and IPv4 ACL-based rate-limiting
- L2 ACLs and L2 ACL-based rate-limiting
- PBR
- VLAN ID and Inner VLAN ID translation on the same interface

NOTE
The design of this feature requires that the **qos-tos trust** and **qos-tos mark** commands be used together.

NOTE
You can now directly configure the **qos-tos trust** and **qos-tos mark** commands at the interface-level.

Specifying the trust level

The trust level specifies where you want the device to get the QoS value for a packet received on the interface.

To set the trust level for an interface to IP Precedence, enter the following command at the configuration level for the interface.

```
NetIron(config-if-1/1)# qos-tos trust ip-prec
```

Syntax: [no] qos-tos trust < cos | ip-prec | dscp >

The **cos | ip-prec | dscp** parameter specifies the trust level:

- **cos** – The device uses the IEEE 802.1p (CoS) priority value in the packet's Ethernet frame header. Use this trust option when you plan to mark the packet's DSCP value based on the incoming IEEE 802.1p value.
- **ip-prec** – The device uses the three most-significant bits in the packet's ToS field and interprets them as an IP precedence value. Use this trust option when the incoming packet is from a device that does not support DSCP and you need to mark the packet for QoS on DSCP devices.
- **dscp** – The device uses the six most-significant bits in the packet's ToS field and interprets them as a DSCP value.

Enabling marking

Marking changes the value of an outbound packet's IEEE 802.1p priority field, DSCP field, or both to match the results of the QoS mappings performed by the device. When you enable marking on an interface, the marking applies to packets that enter the device through that interface.

The following example enables marking for traffic that arrives on port 1/1 and enables the **qos pcp encode-policy on** command on egress port 1/14, as shown.

```
NetIron(config-if-e10000-1/1)# qos-tos mark cos
NetIron(config-if-e10000-1/1)# interface ethernet 1/14
NetIron(config-if-e10000-1/1)# qos pcp encode-policy on
```

This command enables marking of the IEEE 802.1p field in the Ethernet frame.

Syntax: [no] qos-tos mark cos | dscp

The **cos | dscp** parameter specifies the type of marking:

- **cos** – The device changes the outbound packet's IEEE 802.1p priority value to match the results of the device's QoS mapping from the specified trust level.
- **dscp** – The device changes the outbound packet's IEEE 802.1p priority value to match the results of the device's QoS mapping from the specified trust level.

NOTE
The **qos pcp encode-policy on** command must be configured when the **qos-tos mark cos** command is configured. The **qos pcp encode-policy** command is on by default and does not require explicit configuration unless it has been configured to be **off**.

NOTE
You can't apply an ACL to an interface in the outbound direction to change the priority of certain types of traffic.

Packet mapping commands

The **qos-tos trust** command specifies that a COS, IP-Precedence, or DSCP value received on an Ingress port will be used to determine the QoS value that is marked on an outgoing packet on an egress port. The **qos-tos mark** command directs the router to mark outgoing packets with a COS or DSCP value as specified in the command. The value to be marked is determined by a mapping between the value received on the Ingress port and another value that you set using one of the following procedures:

- Changing the CoS -> DSCP Mappings
- Changing the IP Precedence –> DSCP Mappings
- Changing the DSCP –> DSCP Mappings

Changing the CoS –> DSCP mappings

The CoS –> DSCP mappings are used if the trust level is CoS as set by the **qos-tos trust** command.

To change the CoS –> DSCP mappings, enter commands such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# qos-tos map cos-dscp 0 33 25 49 17 7 55 41
NetIron(config)# ip rebind-acl all
```

This command configures the mappings displayed in the COS-DSCP map portion of the QoS information display.

```
NetIron(config-if-1/1)# show qos-tos
```

...portions of table omitted for simplicity...

```
COS-DSCP map:

     COS: 0 1 2 3 4 5 6 7
  ------------------------------------------------------
     dscp: 0 33 25 49 17 7 55 41
```

Syntax: **[no] qos-tos cos-dscp** <*dscp1*> <*dscp2*> <*dscp3*> <*dscp4*> <*dscp5*> <*dscp6*> <*dscp7*> <*dscp8*>

The <*dscp1*> ... <*dscp8*> parameters specify the DSCP values you are mapping to the eight CoS values. You must enter DSCP values for all eight CoS values, in order from CoS value 0 – 7.

NOTE
To place a qos-tos mapping change into effect, you must enter the **ip rebind-acl all** command at the global CONFIG level of the CLI after making the mapping change. This applies to mappings that are configured using the **qos-tos map** command.

Changing the IP precedence –> DSCP mappings

The IP precedence –> DSCP mappings are used if the trust level is IP Precedence as set by the **qos-tos trust** command.

To change the IP precedence –> DSCP mappings, enter commands such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# qos-tos map ip-prec-dscp 0 32 24 48 16 8 56 40
NetIron(config)# ip rebind-acl all
```

This command configures the mappings displayed in the IP Precedence-DSCP map portion of the QoS information display.

```
NetIron(config-if-1/1)# show qos-tos
```

...portions of table omitted for simplicity...

```
IP Precedence-DSCP map:

  ip-prec:     0     1     2     3     4     5     6     7
  ---------------------------------------------------------
     dscp:     0    32    24    48    16     8    56    40
```

For information about the rest of this display, refer to "Displaying QoS configuration information".

Syntax: **[no] qos-tos map ip-prec-dscp** <*dscp1*> <*dscp2*> <*dscp3*> <*dscp4*> <*dscp5*> <*dscp6*> <*dscp7*> <*dscp8*>

The <*dscp1*> ... <*dscp8*> parameters specify the DSCP values you are mapping to the IP precedence values. You must enter DSCP values for all eight IP precedence values, in order from IP precedence value 0 – 7.

NOTE
To place a qos-tos mapping change into effect, you must enter the **ip rebind-acl all** command at the global CONFIG level of the CLI after making the mapping change. This applies to mappings that are configured using the **qos-tos map** command.

Changing the DSCP –> DSCP mappings

To change a DSCP –> DSCP mapping, enter a command such as the following at the global CONFIG CLI level.

```
NetIron(config)# qos-tos map dscp-dscp 0 10
NetIron(config)# ip rebind-acl all
```

This command changes the mapping of DSCP value 0 from 0 to 10.

Syntax: [no] qos-tos map dscp-dscp <*old-dscp-value*> [<*old-dscp-value*>...]
to <*new-dscp-value*> [<*new-dscp-value*>...]

You can change up to eight DSCP values in the same commend. Make sure you enter the old values and their new values in the same order.

NOTE
To place a qos-tos mapping change into effect, you must enter the **ip rebind-acl all** command at the global CONFIG level of the CLI after making the mapping change. This applies to mappings that are configured using the **qos-tos map** command.

Configuring support for super aggregate VLANs

In a super-aggregate VLAN application, you can optionally configure an untagged interface to copy the QOS bits from the tag value set by the edge device to the tag value set by the core device. This is only supported if the incoming packet has ETYPE 0x8100. This can be configured using the **qos decode-cvlan-pcp** command as shown in the following.

```
NetIron(config)# interface ethernet 10/1
NetIron(config-if-e10000-10/1)qos decode-cvlan-pcp
```

Syntax: [no] qos decode-cvlan-pcp

NOTE
The command **aggregated-vlan-copy-cos** is available at the physical interface level to copy the COS value from the internal to the external VLAN tag (for SAV). This command will be automatically migrated to the new command **qos decode-cvlan-pcp**.

Configuring port-level QoS commands on LAG ports

When applying port-level QoS commands to ports in a LAG, the rules can differ according the following:

- For port-level QoS Configurations where QoS Values are Applied Directly to the Port. These commands include the followings: **priority**, **priority force**, **drop-precedence**, **drop-precedence force**.
- For Port-level QoS configurations using commands that begin with the **qos** keyword. These commands include: **qos use-dei, qos dscp decode-policy**, **qos pcp decode-policy**, **qos exp decode-policy**, **qos dscp force**, **qos pcp force**, **qos exp force**, **qos dscp encode-policy**, **qos pcp encode-policy**, and **qos exp encode-policy**.

LAG configuration rules where QoS values are applied directly to the port

In port-level QoS Configurations where QoS values are applied directly to the port, the considerations listed below must be followed.

1. Each port that is configured into the LAG, must have the same **priority**, **priority force**, **drop-precedence**, and **drop-precedence force** configuration.

   If you try to configure a LAG with ports that have a different configuration for these commands, the LAG deployment will fail and you will get an error message as shown in the following.

```
NetIron(config)# lag mylag static
NetIron(config-lag-mylag)# ports eth 10/1 to 10/2
NetIron(config-lag-mylag)# primary 10/1
NetIron(config-lag-mylag)# deploy
port 10/1 priority is 5, but port 10/2 priority is 0
Error: port 10/1 and port 10/2 have different configurations
LAG mylag deployment failed!
NetIron(config-lag-mylag)#
```

2. If you have already formed a LAG with the same configuration, you can change the configuration by making changes to the LAG's primary port.
3. If the LAG configuration is deleted, each of the port in the LAG (primary and secondary) will inherit the QoS configuration of the primary port.

LAG configuration rules for QoS configurations using commands that begin with the qos keyword

In port-level QoS Configurations where QoS Configurations Using Commands that begin with the **qos** keyword are used, the considerations listed below must be followed.

1. The secondary ports configured in the LAG must not have any QoS values configured on them.
2. The **qos** commands that are configured on the primary port are applied to all ports in the LAG.
3. After the LAG is formed, you can change the QoS configuration for all ports in the LAG by making changes to the LAG's primary port, but you cannot change the QoS configurations directly on any of the secondary ports.
4. If the LAG is deleted, the QoS configuration will only be retained on the primary port.

Displaying QoS information

You can display the following QoS information as described:

- **QoS Configuration Information** – Using the **show qos-map decode-map** and **show qos-map encode-map** commands, you can display the priority and drop-precedence values mapped between values internal to the router and values that are received at the router or marked on packets leaving the router. This is described in "Displaying QoS configuration information" on page 321.
- **QoS Packet and Byte Statistics** – Using the **show qos-map decode-map** and **show qos-map encode-map** commands, you can enable and display the contents of the QoS Packet and Byte Counters as described in "Displaying QoS packet and byte counters" on page 324.

Displaying QoS configuration information

You can display the following QoS Configuration information:

- QoS Decode Policy Map Configurations
- QoS Policy Map Binding Configurations

Displaying QoS Decode Policy Map configurations

To display QoS Decode Policy Map configuration information, enter the following command at any level of the CLI.

```
NetIron(config)# Show qos-map dscp decode-map test1
DSCP decode map test1
  DSCP  0 to priority  0 drop-precedence  0
  DSCP  1 to priority  0 drop-precedence  0
  DSCP  2 to priority  0 drop-precedence  1
  DSCP  3 to priority  0 drop-precedence  1
  DSCP  4 to priority  0 drop-precedence  2
  DSCP  5 to priority  0 drop-precedence  2
  DSCP  6 to priority  0 drop-precedence  3
  DSCP  7 to priority  0 drop-precedence  3
  DSCP  8 to priority  7 drop-precedence  0
  DSCP  9 to priority  1 drop-precedence  0
  DSCP 10 to priority  6 drop-precedence  1
  DSCP 11 to priority  1 drop-precedence  1
  DSCP 12 to priority  1 drop-precedence  2
  DSCP 13 to priority  1 drop-precedence  2
  DSCP 14 to priority  1 drop-precedence  3
  DSCP 15 to priority  1 drop-precedence  3
  DSCP 16 to priority  2 drop-precedence  0
  DSCP 17 to priority  2 drop-precedence  0
  DSCP 18 to priority  2 drop-precedence  1
  DSCP 19 to priority  2 drop-precedence  1
  DSCP 20 to priority  2 drop-precedence  2
  DSCP 21 to priority  7 drop-precedence  2
  DSCP 22 to priority  2 drop-precedence  3
  DSCP 23 to priority  2 drop-precedence  3
  DSCP 24 to priority  3 drop-precedence  0
  DSCP 25 to priority  3 drop-precedence  0
  DSCP 26 to priority  3 drop-precedence  1
  DSCP 27 to priority  3 drop-precedence  1
  DSCP 28 to priority  3 drop-precedence  2
  DSCP 29 to priority  3 drop-precedence  2
  DSCP 30 to priority  2 drop-precedence  1
  DSCP 31 to priority  3 drop-precedence  3
   ....
```

Syntax: show qos-map dscp | exp | pcp decode-map <*map-name*> **| all-zero-map | default-map**

The **dscp** option is used to display an Ingress DSCP Policy Map configuration.

The **exp** option is used to display an Ingress EXP Policy Map configuration.

The **pcp** option is used to display an Ingress PCP Policy Map configuration.

The <*map-name*> variable is the name of the Ingress Policy Map whose configuration you want to display.

The **all-zero-map** option is used to display the specified Ingress Policy Map's all-zero-map configuration.

The **default-map** option is used to display the specified Ingress Policy Map's default configuration.

Displaying QoS Egress Encode Policy Map configurations

To display QoS Egress Encode Policy Map configuration information, enter the following command at any level of the CLI.

```
NetIron(config)# show qos-map dscp encode-map test2
DSCP encode map test2
  Priority  0 drop-precedence  0 to DSCP  0
  Priority  0 drop-precedence  1 to DSCP  2
  Priority  0 drop-precedence  2 to DSCP  4
  Priority  0 drop-precedence  3 to DSCP  6
  Priority  1 drop-precedence  0 to DSCP 44
  Priority  1 drop-precedence  1 to DSCP 44
  Priority  1 drop-precedence  2 to DSCP 44
  Priority  1 drop-precedence  3 to DSCP 44
  Priority  2 drop-precedence  0 to DSCP 20
  Priority  2 drop-precedence  1 to DSCP 25
  Priority  2 drop-precedence  2 to DSCP 20
  Priority  2 drop-precedence  3 to DSCP 20
  Priority  3 drop-precedence  0 to DSCP 55
  Priority  3 drop-precedence  1 to DSCP 55
  Priority  3 drop-precedence  2 to DSCP 55
  Priority  3 drop-precedence  3 to DSCP 55
  Priority  4 drop-precedence  0 to DSCP 32
  Priority  4 drop-precedence  1 to DSCP 34
  Priority  4 drop-precedence  2 to DSCP 36
  Priority  4 drop-precedence  3 to DSCP 38
  Priority  5 drop-precedence  0 to DSCP 54
  Priority  5 drop-precedence  1 to DSCP 54
  Priority  5 drop-precedence  2 to DSCP 54
  Priority  5 drop-precedence  3 to DSCP 54
  Priority  6 drop-precedence  0 to DSCP 48
  Priority  6 drop-precedence  1 to DSCP 50
  Priority  6 drop-precedence  2 to DSCP 52
  Priority  6 drop-precedence  3 to DSCP 54
  Priority  7 drop-precedence  0 to DSCP 27
  Priority  7 drop-precedence  1 to DSCP 27
  Priority  7 drop-precedence  2 to DSCP 27
  Priority  7 drop-precedence  3 to DSCP 27
```

Syntax: show qos-map dscp | exp | pcp encode-map <*map-name*> **| all-zero-map | default-map**

The **dscp** option is used to display an Egress DSCP Policy Map configuration.

The **exp** option is used to display an Egress EXP Policy Map configuration.

The **pcp** option is used to display an Egress PCP Policy Map configuration.

The <*map-name*> variable is the name of the Egress Policy Map whose configuration you want to display.

The **all-zero-map** option is used to display the specified Egress Policy Map's all-zero-map configuration.

The **default-map** option is used to display the specified Egress Policy Map's default configuration.

Displaying QoS Binding configurations

To display QoS Binding configuration information, enter the following command at any level of the CLI.

```
NetIron(config)# show qos-map binding global
 qos pcp decode-policy pcp-t2
 qos exp decode-policy exp-t1
 qos dscp decode-policy dscp-t3
 qos dscp encode-policy dscp-d3
```

Syntax: show qos-map binding global | *<slot/port>*

The **global** option is used to display all QoS Policy Map bindings configured on the router.

The *<slot/port>* variable is used to display all QoS Policy Map bindings configured on the router.

Displaying QoS packet and byte counters

You can enable and display the collection of statistics for Ingress and Egress packet priorities as described in the following sections:

- Enabling QoS Packet and Byte Counters
- Displaying QoS Packet and Byte Counters
- Clearing QoS Packet and Byte Counters

Enabling QoS packet and byte counters

You can enable the collection of statistics for Ingress and Egress packet priorities using the **enable-qos-statistics** command as shown in the following.

```
NetIron# enable-qos-statistics
NetIron#
```

Syntax: [no] enable-qos-statistics

The default for this command is disabled.

Using the **no** option returns a previous enabled configuration to the default disabled state.

Displaying QoS packet and byte counters

You can enable the collection of statistics for Ingress and Egress packet priorities using the **enable-qos-statistics** command. Once the collection of statistics is enabled, the **show np qos statistics** command can be used to display a count of the packet priorities of Ingress and Egress packets as shown in the following.

```
NetIron# show np qos statistics eth 1/1
Port 1/1
  Ingress counters:
    COS 0: packets 0                    bytes 0
    COS 1: packets 0                    bytes 0
    COS 2: packets 0                    bytes 0
    COS 3: packets 0                    bytes 0
    COS 4: packets 0                    bytes 0
    COS 5: packets 0                    bytes 0
    COS 6: packets 0                    bytes 0
    COS 7: packets 1122084909           bytes 134650189080
  Egress counters:
    COS 0: packets 0                    bytes 0
    COS 1: packets 0                    bytes 0
    COS 2: packets 0                    bytes 0
    COS 3: packets 0                    bytes 0
    COS 4: packets 4056756685           bytes 486810801752
    COS 5: packets 0                    bytes 0
    COS 6: packets 0                    bytes 0
    COS 7: packets 453                  bytes 49490
```

Syntax: show np qos statistics ethernet <*slot/port*> **| pos** <*slot/port*> **| slot** <*slot-number*>

The **ethernet** option is used to display all QoS counters for the ethernet interface specified by the <*slot/port*> variable.

The **pos** option is used to display all QoS counters for the packet-over-sonet interface specified by the <*slot/port*> variable.

The **slot** option is used to display all QoS counters for the interface module whose location is specified by the <*slot-number*> variable.

Table 61 describes the parameters displayed for the show np qos statistics command.

TABLE 61 QoS counter information

This field...	Displays...
Ingress Counters	Statistics displayed below this heading are for packets arriving on the Ingress port (or ports) before any overrides or merging of packet priorities have been performed.
COS <*num*>: packets	The number of packets that have arrived on the specified port or module with a DSCP, EXP, or PCP value equal to the value of the <*num*> variable.
COS <*num*>: bytes	The number of bytes contained in the packets that have arrived on the specified port or module with a DSCP, EXP, or PCP value equal to the value of the <*num*> variable.
Egress Counters	Statistics displayed below this heading are for packets leaving the router on the Egress port (or ports) accounting for all priority modifications that have been performed on them. These statistics accurately reflect the values for packets that are forwarded out of the router.

TABLE 61 QoS counter information (Continued)

This field...	Displays...
COS <*num*>: packets	The number of packets leaving the router on the specified port or module with a DSCP, EXP, or PCP value equal to the value of the <*num*> variable.
COS <*num*>: bytes	The number of bytes contained in the packets leaving the router on the specified port or module with a DSCP, EXP, or PCP value equal to the value of the <*num*> variable.

Clearing the QoS packet and byte counters

You can clear the QoS counters whose display is generated using the **show np qos statistics** command as shown in the following.

Syntax: clear np qos statistics ethernet <*slot/port*> **| pos** <*slot/port*>

The **ethernet** option is used to clear all QoS counters for the ethernet interface specified by the <*slot/port*> variable.

The **pos** option is used to clear all QoS counters for the packet-over-sonet interface specified by the <*slot/port*> variable.

Weighted Random Early Discard (WRED)

On the PowerConnect Router, queues are provided to buffer traffic levels that exceed the bandwidth of individual ports. For each output port, a set of eight priority queues is allocated on each inbound traffic manager. When traffic exceeds the bandwidth of a port, packets are dropped randomly as long as the congestion persists. Under these conditions, traffic of greater priority can be dropped instead of traffic with a lesser priority.

Instead of being subject to this random process, you can configure a PowerConnect Router to monitor traffic congestion and drop packets according to a WRED (Weighted Random Early Discard) algorithm. This algorithm enables the system to detect the onset of congestion and take corrective action. In practice, WRED causes a router to start dropping packets as traffic in the router starts to back up. WRED provides various control points that can be configured to change a system's reaction to congestion. The following variables are used when calculating whether to drop or forward packets:

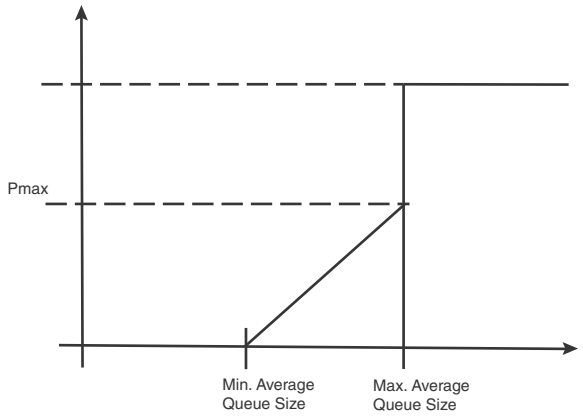
- **Statistical Average-Q-Size** – The statistical average size of the queue calculated over time on the router.
- **Current-Q-Size** – The current size of the queue as calculated on the router.
- **Wq** – This variable specifies the weights that should be given to the current queue size and the statistical average-q-size when calculating the size for WRED calculations.
- **Max-Instantaneous-Q-Size** – The maximum size up to which a queue is allowed to grow. Packets that cause the queue to grow beyond this point are unconditionally dropped. This variable is user configured.
- **Min-Average-Q-Size** – The average queue size below which all packets are accepted. This variable is user configured.
- **Max-Average-Q-Size** – The average queue size above which all packets are dropped. This variable is user configured.

- **Pmax** – The maximum drop probability when queue-size is at Max-Average-Q-Size. This variable is user configured.
- **Pkt-Size-Max** – The packet size to which the current packet's size is compared as shown in the algorithm below. This variable is user configured.

How the WRED algorithm operates

The graph in Figure 18 describes the interaction of the previously described variables in the operation of WRED. When a packet arrives at a router, the average queue size (**q-size**) is calculated (note that this is not the statistical average queue size - refer to "Calculating avg-q-size" on page 327). If **q-size** as calculated is below the configured Min. Average Queue Size, then the packet is accepted. If the average queue size is above the Max. configured Average Queue Size threshold, the packet is dropped. If the instantaneous queue size exceeds the value configured for the Max-Instantaneous-Q-Size, the packet is dropped. If the Average Queue size falls between the Min. Average Queue Size and the Max. Average Queue Size, packets are dropped according to the calculated probability described in "Calculating packets that are dropped" on page 328.

FIGURE 18 WRED operation graph



Calculating avg-q-size

The algorithm first calculates the **avg-q-size** through the following equation.

avg-q-size = ((1 - Wq) * Statistical Average-Q-Size) + (Wq * Current-Q-Size)

The user-configured **Wq** value is instrumental to the calculation and can be:

- equal to the statistical average queue size (**Wq** == 0), or
- equal to the current queue size (**Wq** == 1) or
- be between 0 and 1 (0 < **Wq** < 1).

Lower Wq values cause the **avg-q-size** to lean towards the statistical average queue size, reducing WRED's sensitivity to the current state of the queue and thus reducing WRED's effectiveness. On the other hand, higher **Wq** values cause the **avg-q-size** to lean towards the instantaneous queue size, which exposes WRED to any change in the instantaneous queue size and thus may cause WRED to overreact in cases of bursts. Thus, the value of **Wq** should be carefully chosen according to the application at hand.

Calculating packets that are dropped

The **Pdrop** value, as calculated in the following equation, is the probability that a packet will be dropped in a congested router.

$$Pdrop = \frac{\text{pkt-size}}{\text{pkt-size-max}} * Pmax * \frac{\text{(avg-q-size - min-avg-q size)}}{\text{(max-avg-q-size - min-avg-q size)}}$$

Applying the WRED algorithm to router traffic

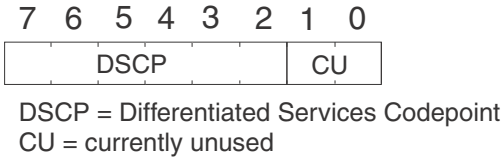
Packets are assigned to an Ingress queue type based on their individual destination port and one of the 8 (0 - 7) internal priorities. Each of these priorities is assigned a queue type from 0 - 7 according to the internal priority it belongs to as shown in Table 62.

TABLE 62 Internal priority to queue type mapping

Internal priority	0	1	2	3	4	5	6	7
Queue type	0	1	2	3	4	5	6	7

The WRED algorithm is applied to traffic on these individual queues based upon parameters configured for its assigned queue type. When traffic arrives at a queue, it is passed or dropped as determined by the WRED algorithm. Packets in an individual queue are further differentiated by one of four drop precedence values which are determined by the value of bits 3:2 of the TOS or DSCP bits in the IPv4 or IPv6 packet header as shown in Figure 19.

FIGURE 19 TOS or DSCP bits in packet header



The user configurable values applied per queue type and per drop precedence value are:

- Maximum Drop Probability
- Minimum and Maximum Average Queue Size
- Maximum Packet Size

Configuring packet drop priority using WRED

For a description of WRED, refer to "Weighted Random Early Discard (WRED)" on page 326. This section describes how to configure the parameters described in that section to enable the use of WRED on a PowerConnect router. In addition, there is a default configuration that can be enabled that sets the parameters to the values shown in Table 64. If you use the default configuration, you do not need to set the parameters individually.

To configure WRED, you must configure the following parameters:

- "Enabling WRED"
- "Setting the averaging-weight (Wq) parameter" (optional)

- "Configuring the maximum instantaneous queue size" (optional)
- "Configuring the drop precedence parameters" (optional)
- "Restoring default WRED parameters"

Enabling WRED

WRED must be enabled for the queue type of any forwarding queue that you want it to operate on. To enable WRED for the forwarding queues with a queue type of 3, enter the following command.

```
NetIron(config)#qos queue-type 3 wred enable
```

Syntax: [no] qos queue-type <*queue-number*> **wred enable**

The <*queue-type*> variable is the number of the forwarding queue that you want to enable WRED for. There are eight forwarding queues on PowerConnect Routers. They are numbered 0 to 7. Default values are as described in Table 64. You can optionally adjust any of the pre-configured parameters described there.

Setting the averaging-weight (Wq) parameter

The Wq parameter described in "Weighted Random Early Discard (WRED)" on page 326 is configured as the **averaging-weight** parameter. In this implementation, you can set one of 13 (1 - 13) possible values. These values represent a Wq value as described in Table 63

TABLE 63 Possible Wq values

Averaging weight setting	Wq value as a percentage
1	50%
2	25%
3	12.5%
4	6.2%
5	3.12%
6	1.56%
7	0.78%
8	0.4%
9	0.2%
10	0.09%
11	0.05%
12	0.02%
13	0.01%

To set the wq parameter for queues with a queue type of 1 to 25%, use the following command.

```
NetIron(config)#qos queue-type 1 wred averaging-weight 2
```

This gives the current queue size a weight of 25% over the statistical average queue size.

Syntax: [no] qos queue-type <*queue-type*> **wred averaging-weight** <*avg-weight-value*>

The <*queue-type*> variable is the number of the forwarding queue type that you want to configure the **averaging-weight** (Wq) parameter for. There are eight forwarding queue types on PowerConnect Routers. They are numbered 0 to 7.

The <*avg-weight-value*> variable is the weight-ratio between instantaneous and average queue sizes. It is described as the Wq parameter in "Weighted Random Early Discard (WRED)" on page 326. It can be one of the 13 values expressed as 1 to 13 described in Table 63. The default value is 9 which maps to a Wq value of .19%.

Configuring the maximum instantaneous queue size

You can set the maximum size to which a queue is allowed to grow. Packets that cause the queue to grow beyond this setting are unconditionally dropped. To set the maximum instantaneous queue size for queues with a queue type of 1 to 32000, use the following command.

```
NetIron(config)#qos queue-type 1 max-queue-size 32
```

Syntax: [no] qos queue-type <*queue-number*> **max-queue-size** <*max-queue*>

The <*queue-type*> variable is the number of the forwarding queue type that you want to configure the **instantaneous-queue-size** parameter for. There are eight forwarding queue types on PowerConnect Routers. They are numbered 0 to 7.

The <*max-queue*> variable is the maximum size to which a queue is allowed to grow. It is defined in Kbytes. The default values are shown in Table 64.

Configuring the drop precedence parameters

The DSCP or TOS bits in packets are used to prioritize packet delivery for specified queue types. These values are from 0 to 4. Packets with a DSCP or TOS value of 0 are least likely to be dropped and packets with a DSCP or TOS of 3 are most likely to be dropped.

NOTE
In addition to bits in the DSCP, the DP option can use other fields (in the PCP header or the EXP bit header) to control WRED in the priority queues.

In addition, the maximum drop probability, the minimum and maximum average queue size, and the maximum packet size can be configured to apply selectively to packets with a specified queue type and DSCP or TOS value. The following sections describe how to set the following drop precedence parameters for each of the four DSCP or TOS values for each of the four queue types:

- "Setting the maximum drop probability"
- "Setting the minimum and maximum average queue size"
- "Setting the maximum packet size"

NOTE
Packets that do not have the DSCP or TOS value set are assigned a drop precedence equal to the DSCP or TOS level of 0.

Setting the maximum drop probability

To set the maximum drop probability for queue type 1 and drop precedence 0 when the queue size reaches the Max-average-q-size value to 20% use the following command.

```
NetIron(config)#qos queue-type 1 wred drop-precedence 0 drop-probability-max 20
```

Syntax: **[no] qos queue-type** <*queue-type*> **wred drop-precedence** <*drop-precedence-value*> **drop-probability-max** <*p-max*>

The <*queue-type*> variable is the number of the forwarding queue type that you want to configure drop-precedence for. There are eight forwarding queue types on PowerConnect Routers. They are numbered 0 to 7.

The <*drop-precedence-value*> variable for the drop-precedence parameter is the TOS or DSCP value in the IPv4 or IPv6 packet header. It determines drop precedence on a scale from **0 - 3**. Packets than contain a DSCP value of **0** are least likely to be dropped and packets with a value of **3** are most likely to be dropped. The default value is **0**.

The <*p-max*> variable defines the maximum drop probability when the queue size is at the value configured for **max-avg-q-size**. This value is expressed as a percentage. The default values are shown in Table 64.

Setting the minimum and maximum average queue size

Configuration Considerations

When setting the minimum and maximum average queue size, consider the following:

- If a user enters a min-avg-queue-size value that is equal to what is currently configured for the max-avg-queue-size, then the min-avg-queue-size is decremented by 64. The min-avg-queue-size is decremented by 64 because the value must be different from the max-avg-queue-size that is currently configured. The following example is a warning message that is displayed on the console:

```
Warning: The min-avg-queue-size is decreased to(min-avq-queue-size-64)as min and
max should be different to be effective.
```

- If a user enters a max-avg-queue-size value that is equal to what is currently configured for the min-avg-queue-size, then the max-avg-queue-size is incremented by 64. The max-avg-queue-size is incremented by 64 because the value must be different from the min-avg-queue-size that is currently configured. The following example is a warning message that is displayed on the console:

```
Warning: The max-avg-queue-size is increased to(max-avq-queue-size+64)as min and
max should be different to be effective.
```

- If a user enters a min-avg-queue-size value equal to the max-avg-queue-size, then the max-avg-queue-size is incremented by 64. The max-avg-queue-size is incremented by 64 because the value must be different from the min-avg-queue-size. The following example is a warning message that is displayed on the console:

```
Warning: The max-avg-queue-size is increased to(max-avq-queue-size+64)as min and
max should be different to be effective.
```

However if a user enters a max-avg-queue-size and min-avg-queue-size equal to 32768, then the min-avg-queue-size is decremented.

To set the maximum average queue size for queue type 1 and drop precedence 0 to the maximum size of 32768 Kbytes, use the following command.

```
NetIron(config)#qos queue-type 1 wred drop-precedence 0 max-avg-queue-size 32768
```

Syntax: **[no] qos queue-type** <*queue-type*> **wred drop-precedence** <*drop-precedence-value*> **max-avg-queue-size** <*max-size*>

To set the minimum average queue size to the maximum size of 16 Kbytes, use the following command.

```
NetIron(config)#qos queue-type 1 wred drop-precedence 0 min-avg-queue-size 16
```

Syntax: **[no] qos queue-type** <*queue-type*> **wred drop-precedence** <*drop-precedence-value*> **min-avg-queue-size** <*min-size*>

The <*queue-type*> variable is the number of the forwarding queue type that you want to configure drop-precedence for. There are eight forwarding queue types on PowerConnect Routers. They are numbered **0** to **7**.

The <*drop-precedence-value*> variable for the drop-precedence parameter is the TOS or DSCP value in the IPv4 or IPv6 packet header. It determines drop precedence on a scale from **0 - 3**. Packets than contain a DSCP value of **0** are least likely to be dropped and packets with a value of **3** are most likely to be dropped. The default value is **0**.

The <*min-size*> variable is the average queue size below which all packets are accepted. Possible values are 1 - 32768 KBytes. It must be set in multiples of 64K. The default values are shown in Table 64.

The <*max-size*> variable is the average queue size above which all packets are dropped. (1 - 32768) (KBytes) in multiples of 64K. The default values are shown in Table 64.

Setting the maximum packet size

To set the maximum packet size to 16 bytes for queue type 1 and drop precedence 0, use the following command.

```
NetIron(config)#qos queue-type 1 wred drop-precedence 0 packet-size-max 16
```

Syntax: **[no] qos queue-type** <*queue-type*> **wred drop-precedence** <*drop-precedence-value*> **packet-size-max** <*pkt-size*>

The <*queue-type*> variable is the number of the forwarding queue type that you want to configure drop-precedence for. There are eight forwarding queue types on PowerConnect Routers. They are numbered 0 to 7.

The <*drop-precedence-value*> variable for the drop-precedence parameter is the TOS or DSCP value in the IPv4 or IPv6 packet header. It determines drop precedence on a scale from 0 - 3. Packets than contain a DSCP value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped. The default value is 0.

The <*pkt-size*> variable is the pkt-size-max variable used in the equation described in "Calculating packets that are dropped" on page 328. Permissible values are an even number of bytes between 16 and 32768. The default values are shown for each queue type and drop precedence value in Table 64.

Restoring default WRED parameters

Table 64 describes all of the default values for each of the WRED parameters. If you change any of the values from the default values, you can restore the defaults per queue type. To reset the queue type 1 with default values for the WRED parameters, use the following command.

```
NetIron(config)#qos queue-type 1 wred default-params
```

Syntax: **[no] qos queue-type** <*queue-number*> **default-params**

The <*queue-number*> variable is the number of the forwarding queue that you want to configure drop-precedence for. There are eight forwarding queues on PowerConnect Routers. They are numbered 0 to 7

TABLE 64 WRED default settings

Queue type	Drop precedence	Minimum average queue size (KByte)	Maximum average queue size (KByte)	Maximum packet size (Byte)	Maximum drop probability	Maximum instantaneous queue size (Kbyte)	Average weight
0	0	320	1024	16384	2%	1024	6.25%
	1	256	1024	16384	4%		
	2	256	1024	16384	9%		
	3	192	1024	16384	10%		
1	0	320	1024	16384	2%	1024	6.25%
	1	256	1024	16384	4%		
	2	256	1024	16384	9%		
	3	192	1024	16384	9%		
2	0	384	1024	16384	2%	1024	6.25%
	1	320	1024	16384	4%		
	2	256	1024	16384	9%		
	3	256	1024	16384	9%		
3	0	384	1024	16384	2%	1024	6.25%
	1	320	1024	16384	4%		
	2	256	1024	16384	9%		
	3	256	1024	16384	9%		
4	0	384	1024	16384	2%	1024	6.25%
	1	320	1024	16384	4%		
	2	256	1024	16384	9%		
	3	256	1024	16384	9%		
5	0	384	1024	16384	2%	1024	6.25%
	1	320	1024	16384	4%		
	2	256	1024	16384	9%		
	3	256	1024	16384	9%		
6	0	1024	1088	16384	0%	1024	6.25%
	1	448	832	16384	2%		
	2	384	832	16384	5%		
	3	320	832	16384	6%		
7	0	1024	1088	16384	0%	1024	6.25%
	1	448	832	16384	2%		
	2	384	832	16384	5%		
	3	320	832	16384	6%		

NOTES:

- If you enter the **min-avg-queue-size** equal to what is already configured as the **max-avg-queue-size**, then the **min-avg-queue-size** will be decremented by 64 to make it different from the **max-avg-queue-size**, the following warning is displayed: "Warning - **min-avg-queue-size** is decreased to (**min-avg-queue-size** - 64) as min and max should be different to be effective."
- If you enter the **max-avg-queue-size** equal to what is already configured as the min-avg-queue-size, then the **max-avg-queue-size** will be incremented by 64 to make it different from the **min-avg-queue-size**, the following warning is displayed: "Warning - max-avg-queue-size is increased to (max-avg-queue-size + 64) as the min & max should be different to be effective."
- If you enter the min-average-queue-size equal to the **max-avg-queue-size**, the **max-avg-queue-size** will be incremented by 64 to make it different from **min-avg-queue-size**, the following warning is displayed: "Warning - **max-avg-queue-size** incresed to (**max-avg-queue-size + 64**) as min & max should be different to be effective." Unless you enter the **max-avg-queue-size** and **min-avg-queue-size** equal to 32768, the **min-avg-queue-size** will be decremented.

Displaying the WRED configuration

To view a WRED configuration, use the following command.

```
NetIron# show qos wred
QType Enable AverWt    MaxQSz DropPrec MinAvgQSz MaxAvgQSz MaxDropProb MaxPktSz
0     Yes    9(0.19%)  16384  0            5696     16384          2%      16384
                               1            4864     16384          4%      16384
                               2            4096     16384          9%      16384
                               3            3264     16384         10%      16384
1     No
2     No
3     Yes    9(0.19%)  16384  0            6528     16384          2%      16384
                               1            5696     16384          4%      16384
                               2            4864     16384          9%      16384
                               3            4096     16384          9%      16384
4     No
5     No
6     No
7     No
```

Scheduling traffic for forwarding

If the traffic being processed by a PowerConnect is within the capacity of the router, all traffic is forwarded as received. Once we reach the point where the router is bandwidth constrained, it becomes subject to drop priority if configured as described in "Configuring packet drop priority using WRED" on page 328 or traffic scheduling as described in this section.

The PowerConnect routers classify packets into one of eight internal priorities. Traffic scheduling allows you to selectively forward traffic according to the forwarding queue that is mapped to according to one of the following schemes:

- **Strict priority-based scheduling** – This scheme guarantees that higher-priority traffic is always serviced before lower priority traffic. The disadvantage of strict priority-based scheduling is that lower-priority traffic can be starved of any access.

- **WFQ weight-based traffic scheduling** – With WFQ destination-based scheduling enabled, some weight-based bandwidth is allocated to all queues. With this scheme, the configured weight distribution is guaranteed across all traffic leaving an egress port and an input port is guaranteed allocation in relationship to the configured weight distribution.
- **Mixed strict priority and weight-based scheduling** – This scheme provides a mixture of strict priority for the three highest priority queues and WFQ for the remaining priority queues.

Configuring traffic scheduling

Traffic scheduling can be configured on a per-port basis. It affects the outgoing traffic on the configured port when bandwidth congestion occurs on that port. The following sections describe how to configure each of the traffic scheduling schemes:

- "Configuring strict priority-based traffic scheduling" This option is the default traffic scheduling method if traffic scheduling is not configured on a port.
- "Calculating the values for WFQ Weight-based traffic scheduling"
- "Configuring WFQ weight-based traffic scheduling"
- "Configuring mixed strict priority and weight-based scheduling"

Configuring strict priority-based traffic scheduling

To configure strict priority-based scheduling use a command such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# qos scheduler strict
```

Syntax: qos scheduler strict

This is the default when traffic scheduling is not configured.

Calculating the values for WFQ Weight-based traffic scheduling

Weighted Fair Queueing (WFQ) scheduling is configured to be a percentage of available bandwidth using the following formula.

$$\text{Weight of } q(x) = \frac{q(x)}{q0 + q1 + q2 + q3 + q4 + q5 + q6 + q7}$$

Where

q (x) = The value of the queue that you want to determine the weight for. It can be the value of any queue (0 - 7).

q0 - q7 = the assigned values of the eight queues.

Weight of q (x) = the calculated weight as a percentage of the port's total bandwidth.

For example if you assign the following values to queues 0 to 7:

- Queue 0 =10, Queue 1 = 15, Queue 2 = 20, Queue 3 = 25, Queue 4 = 30, Queue 5 = 35, Queue 6 = 40, and Queue 7 = 45,

To determine the weight of **q3**, use the following formula.

$$\text{Weight of q3} = \frac{25}{10 + 15 + 20 + 25 + 30 + 35 + 40 + 45}$$

The weight of q3 is 11.4%. Consequently, q3 will get 11.4% of the port's total bandwidth.

The values of the remaining queues are calculated to be the following: q7 = 20.5%, q6 = 18.2%, q5 = 15.9%, q4 = 13.6%, q3 = 11.4%, q2 = 9.1%, q1 = 6.8%, and q0 = 4.5%

Configuring WFQ weight-based traffic scheduling

To configure WFQ weight-based scheduling use a command such as the following.

```
NetIron(config)# interface ethernet 1/1
PowerConnect(config-if-e1000-1/1)# qos scheduler weighted 5 10 15 20 30 15 5 10
```

Syntax: **qos scheduler weighted** <*queue7-weight*> <*queue6-weight*> <*queue5-weight*> <*queue4-weight*> <*queue3-weight*> <*queue2-weight*> <*queue1-weight*> <*queue0-weight*>

The <*queue7-weight*> variable defines the relative value for queue7 in calculating queue7's allocated bandwidth.

The <*queue6-weight*> variable defines the relative value for queue6 in calculating queue6's allocated bandwidth.

The <*queue5-weight*> variable defines the relative value for queue5 in calculating queue5's allocated bandwidth.

The <*queue4-weight*> variable defines the relative value for queue4 in calculating queue4's allocated bandwidth.

The <*queue3-weight*> variable defines the relative value for queue3 in calculating queue3's allocated bandwidth.

The <*queue2-weight*> variable defines the relative value for queue2 in calculating queue2's allocated bandwidth.

The <*queue1-weight*> variable defines the relative value for queue1 in calculating queue1's allocated bandwidth.

The <*queue0-weight*> variable defines the relative value for queue0 in calculating queue0's allocated bandwidth

The acceptable range for <*queuex-weight*> variables is 1-128.

Refer to "Calculating the values for WFQ Weight-based traffic scheduling" for information on assigning queue0-weight to queue7-weight values.

Configuring mixed strict priority and weight-based scheduling

When configuring the mixed strict priority and weight-based scheduling option, queues 5 - 7 are allocated to strict priority-based scheduling and queues 0 - 4 are allocated to weight-based scheduling.

To configure mixed priority and weight-based scheduling use a command such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# qos scheduler mixed 100 80 60 40 20
```

Syntax: qos scheduler mixed *<Queue4-weight> <Queue3-weight> <Queue2-weight> <Queue1-weight> <Queue0-weight>*

The *<queue4-weight>* variable defines the relative value for queue4 in calculating queue4's allocated bandwidth.

The *<queue3-weight>* variable defines the relative value for queue3 in calculating queue3's allocated bandwidth.

The *<queue2-weight>* variable defines the relative value for queue2 in calculating queue2's allocated bandwidth.

The *<queue1-weight>* variable defines the relative value for queue1 in calculating queue1's allocated bandwidth.

The *<queue0-weight>* variable defines the relative value for queue0 in calculating queue0's allocated bandwidth

The acceptable range for *<queuex-weight>* variables is 1-128.

Refer to "Calculating the values for WFQ Weight-based traffic scheduling" for information on assigning queue0-weight to queue4-weight values.

Egress port and priority based rate shaping

Rate shaping is a mechanism to smooth out the variations in traffic above a certain rate. The primary difference between rate shaping and rate limiting is that in rate limiting, traffic exceeding a certain threshold is dropped. In rate shaping, the traffic that exceeds a threshold is buffered so that the output from the buffer follows a more uniform pattern. Rate shaping is useful when burstiness in the source stream needs to be smoothed out and a more uniform traffic flow is expected at the destination.

NOTE
Because excess traffic is buffered, rate shaping must be used with caution. In general, it is not advisable to rate shape delay-sensitive traffic.

PowerConnect support Egress rate shaping. Egress rate shaping is supported per port or for each priority queue on a specified port.

Configuring port-based rate shaping

When setting rate shaping for a port, you can limit the amount of bandwidth available on a port within the limits of the port's rated capacity. Within that capacity, you can set the bandwidth at increments within the ranges described in Table 65.

TABLE 65 Port-based rate shaping interval table

Range	Increment supported within the range
0 - 10M	8,333
10M - < 100M	20,833
100 M - < 1G	208,333
1G - 10G	2,083,333

NOTE
The egress rate shaping burst size for a port-based shaper is 10000 bytes.

These limits provide a minimum and maximum rate that the port can be set to. They also provide the increments at which the port capacity can be set. In operation, you can set any number between the minimum and maximum values. The router will automatically round-up the value to the next higher increment.

For example, if you set the rate of a 10G port to 2,000,000,000, the actual rate would be 2,002,083,173. This is because it is the next highest increment above 2,000,000,000.

To set a 10 Gbps port to the incremental port capacity over 2 Gbps, use the following command.

```
NetIron(config)# interface ethernet 2/2
NetIron(config-if-e10000-2/2)# qos shaper 2000000
```

Syntax: [no] qos shaper <*rate*>

The <*rate*> variable sets the rate you want to set for the port within the limits available as described in Table 65.

Configuring port and priority-based rate shaping

When setting rate shaping for a priority queue, you can limit the amount of bandwidth available for a specified priority within the limits of the capacity of the port that the priority is configured on. You can set the limit for the priority to any value from one to the port's maximum rating and the router will automatically round-up the value to the next increment supported. This will be a slightly higher value than what you specify with the command. For example, if you set the rate for priority 2 on a 10G port to 2,000,000,100, the actual rate would be slightly higher.

NOTE
The egress rate shaping burst size for a port and priority-based shaper is 3072 bytes.

To set the capacity for priority 2 traffic on a 10 Gbps port to the incremental capacity over 2 Gbps, use the following command.

```
NetIron(config)# interface ethernet 2/2
NetIron(config-if-e10000-2/2)# qos shaper priority 2 2000000
```

Syntax: [no] qos shaper priority <*priority-level*> <*rate*>

The <*priority-level*> variable specifies the priority that you want to set rate shaping for on the port being configured.

The <*rate*> variable sets the rate you want to set for the priority.

Multicast queue size, flow control and rate shaping

There are four internal priorities for multicast or broadcast traffic. These four priorities are mapped from the router's eight internal priorities as described inTable 66

TABLE 66 Mapping between multicast or broadcast and internal forwarding priorities

Internal Forwarding Priority	0,1	2,3	4,5	6,7
Multicast Internal Priority	0	1	2	3

The internal forwarding priority of a multicast or broadcast packet is determined from the packet's IEEE 802.1p priority, incoming port priority or IP ToS or DSCP as described in the "Default QoS mappings" on page 288. Four multicast queue types (0 to 3) are used for multicast internal priorities 0 to 3 respectively.

Configuring multicast queue size

The following example configures a 2 MByte queue size for queue 0.

```
NetIron(config)# qos multicast-queue-type 0 max-queue-size 2048
```

Syntax: [no] qos multicast-queue-type <*queue-number*> **max-queue-size <queue-size.>**

The <*queue-number*> variable specifies the queue that you want to configure a maximum size for. Possible values are 0 - 3.

The <*queue-size*> variable specifies size in KBytes that you want to set as the maximum value for the specified multicast queue. Possible values are 1 - 32768 KBytes. The default queue size is 1 Mbyte.

This command is applied per router and takes effect on all Traffic Managers within the configured router.

Configuring multicast flow control

Flow controls are available from egress to Ingress, and from fabric to Ingress. At the egress of each Traffic Manager, there are pre-determined thresholds for consumed resources and available resources and separate thresholds for guaranteed multicast or broadcast traffic and best-effort multicast or broadcast traffic. When a threshold is crossed, flow control can be triggered and multicast or broadcast traffic of the corresponding class is stopped at Ingress until resources are below the threshold again. Flow control is disabled by default and can be enabled on an interface using the command shown in the following.

```
NetIron(config)# interface ethernet 2/2
NetIron(config-if-e10000-2/2)# qos multicast flow-control
```

Syntax: [no] qos multicast flow-control

This command changes the flow control setting on the Traffic Manager where the interface resides.

Configuring multicast rate shaping

You can specify either guaranteed or best effort multicast rate shaping for a port in Kilobits per second. Multicast rate shaping is configured per-port to the Ingress port.

The following example changes the best-effort multicast traffic rate to 10 Mbps.

```
NetIron(config)# interface ethernet 2/2
NetIron(config-if-e10000-2/2)# qos multicast shaper best-effort rate 10000
```

Syntax: [no] qos multicast shaper [guaranteed | best-effort] rate <*bandwidth*>

The **guaranteed** option specifies that the multicast or broadcast shaper applies only to internal multicast priority 3 (the highest multicast priority) traffic.

The **best-effort** option specifies that the multicast or broadcast shaper applies to internal multicast priority 0, 1 and 2 traffic only.

The <*bandwidth*> variable specifies the maximum bandwidth in Kbps for best effort or guaranteed multicast traffic scheduled by the Traffic Manager across the switch fabric.

Configuration considerations

When applied to a port, the **qos multicast shaper** configuration is applied to all ports on the Interface module that use the same Traffic Manager as the configured port. This is unlike the behaviour of rate shaping applied for unicast traffic. The relationship between ports and Traffic Managers is defined in the following tables: "Ethernet ports per traffic manager" on page 344.

Example 1

In the following example, multicast rate shaping is applied to port 1/18 on a 20-port, 10/100/1000 Copper Ethernet Interface module.

```
NetIron(config)# interface ethernet 1/18
NetIron(config-if-e1000-1/18)# qos multicast shaper best-effort rate 10000
```

In this example, the configuration will apply to Ingress traffic that arrives on any port of the Interface module.

Example 2

In the following example, multicast rate shaping is applied to port 1/1 of a 4-port, 10 GbE Ethernet Interface module.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# qos multicast shaper best-effort rate 10000
```

In this example, the configuration will apply to Ingress traffic that arrives on either port 1/1 or port 1/2 of the Interface module.

NOTE
When a **qos multicast shaper** command is configured for a port, the configuration command is placed in the running config for all ports that belong to the same Traffic Manager. In Example 1, that would mean that the **qos multicast shaper best-effort rate 10000** command would appear in the interface configuration section for all ports (1 to 20) on the Interface Module. In Example 2, that would mean that the **qos multicast shaper best-effort rate 10000** command would appear in the interface configuration section for ports 1 and 2 on the Interface Module.

Traffic manager statistics display

Counters have been introduced to track the packets and bytes that enter the Ingress traffic manager and exit the egress traffic manager. Data from these counters can be displayed as described in the following sections.

Displaying all traffic manager statistics for a router

The following command displays all traffic manager statistics for a router by port groups that belong to each traffic manager.

```
NetIron# show tm statistics
--------- Ports 2/1 - 2/20 ---------
Ingress Counters:
   Total Ingress Pkt Count:                 464418
   EnQue Pkt Count:                         464418
   EnQue Byte Count:                        51904240
   DeQue Pkt Count:                         464418
   DeQue Byte Count:                        51904240
   TotalQue Discard Pkt Count:              0
   TotalQue Discard Byte Count:             0
   Oldest Discard Pkt Count:                0
   Oldest Discard Byte Count:               0
Egress Counters:
   EnQue Pkt Count:                         701812
   EnQue Byte Count:                        78785888
   Discard Pkt Count:                       0
   Discard Byte Count:                      0
--------- Ports 4/1 - 4/20 ---------
Ingress Counters:
   Total Ingress Pkt Count:                 0
   EnQue Pkt Count:                         0
   EnQue Byte Count:                        0
   DeQue Pkt Count:                         0
   DeQue Byte Count:                        0
   TotalQue Discard Pkt Count:              0
   TotalQue Discard Byte Count:             0
   Oldest Discard Pkt Count:                0
   Oldest Discard Byte Count:               0
Egress Counters:
   EnQue Pkt Count:                         0
   EnQue Byte Count:                        0
   Discard Pkt Count:                       0
   Discard Byte Count:                      0
```

Syntax: show tm statistics

Displaying traffic manager statistics for a port group

The following command displays all traffic manager statistics for a specified port group as identified by a slot and port within the group.

```
NetIron#show tm statistics ethernet 2/1
--------- Ports 2/1 - 2/20 ---------
Ingress Counters:
   Total Ingress Pkt Count:                 464454
   EnQue Pkt Count:                         464454
   EnQue Byte Count:                        51907696
   DeQue Pkt Count:                         464454
   DeQue Byte Count:                        51907696
   TotalQue Discard Pkt Count:              0
   TotalQue Discard Byte Count:             0
   Oldest Discard Pkt Count:                0
   Oldest Discard Byte Count:               0
Egress Counters:
```

```
   EnQue Pkt Count:                          701866
   EnQue Byte Count:                         78791072
   Discard Pkt Count:                        0
   Discard Byte Count:                       0
```

Syntax: show tm statistics ethernet <*slot/port*>

The <*slot/port*> variable specifies the slot and port number of the port group that you want to display traffic manager statistics for.

NOTE
A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68. Specifying a particular port and slot gathers statistics for all ports that belong to the same port group.

Displaying traffic manager statistics for an interface module

The following command displays all traffic manager statistics for an interface module identified by its slot number.

```
NetIron#show tm statistics slot 4
--------- Ports 4/1 - 4/20 ---------
Ingress Counters:
   Total Ingress Pkt Count:                  0
   EnQue Pkt Count:                          0
   EnQue Byte Count:                         0
   DeQue Pkt Count:                          0
   DeQue Byte Count:                         0
   TotalQue Discard Pkt Count:               0
   TotalQue Discard Byte Count:              0
   Oldest Discard Pkt Count:                 0
   Oldest Discard Byte Count:                0
Egress Counters:
   EnQue Pkt Count:                          0
   EnQue Byte Count:                         0
   Discard Pkt Count:                        0
   Discard Byte Count:                       0
```

Syntax: show tm statistics ethernet <*slot/port*>

The **slot** <*slot-number*> option specifies an interface module that you want to display traffic manager statistics from.

TABLE 67 Traffic manager statistics

This field...	Displays...
Ingress Statistics	
Total Ingress Pkt Count	A count of all packets entering into this traffic manager. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
EnQue Pkt Count	A count of all packets entering Ingress queues on this traffic manager. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.

TABLE 67 Traffic manager statistics (Continued)

This field...	Displays...
EnQue Byte Count	A count of all bytes entering Ingress queues on this traffic manager. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
DeQue Pkt Count	A count of all packets dequeued from Ingress queues and forwarded on this traffic manager. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
DeQue Byte Count	A count of all bytes dequeued from Ingress queues and forwarded on this traffic manager.
TotalQue Discard Pkt Count	A count of all packets failing to enter Ingress queues on this traffic manager. This may be due to: • the queue reaching its maximum depth, WRED, or other reasons. • the network processor deciding to drop packets for reasons including: an unknown Layer-3 route, RPF, or segment filtering. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
TotalQue Discard Byte Count	A count of all bytes failing to enter Ingress queues on this traffic manager. This may be due to: • the queue reaching its maximum depth, WRED, or other reasons. • the network processor deciding to drop packets for reasons including: an unknown Layer-3 route, RPF, or segment filtering. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
Oldest Discard Pkt Count	A count of all packets entering Ingress queues on this traffic manager, but deleted afterwards due to buffer full. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
Oldest Discard Byte Count	A count of all bytes entering Ingress queues on this traffic manager, but deleted afterwards due to buffer full. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
Egress statistics	
EnQue Pkt Count	A count of all packets entering egress queues and forwarded out on this traffic manager. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
EnQue Byte Count	A count of all bytes entering egress queues and forwarded out on this traffic manager. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
Discard Pkt Count	A count of all packets failing to enter egress queues on this traffic manager. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.
Discard Byte Count	A count of all bytes failing to enter egress queues on this traffic manager. A traffic manager contains a specific number of ports depending on the Interface module as described in Table 68.

NOTE
The byte counts displayed from the **show tm statistics** command incorporate proprietary internal headers of various lengths.

TABLE 68 Ethernet ports per traffic manager

Interface module	Ports per Traffic Manager (TM)	
	TM 1	TM 2
4 X 10 Gbps (Ethernet)	1 - 2	3 - 4
2 X 10 Gbps (Ethernet)	1 - 2	
20 X 1 Gbps (Ethernet)	1 - 20	

Displaying traffic manager statistics for NI-MLX-10Gx8-M and NI-MLX-10Gx8-D modules

The following command displays traffic manager statistics for the NI-MLX-10Gx8-M module, and the NI-MLX-10Gx8-D module identified by its slot number.

```
PowerConnect#show tm statistics slot 4
--------- Ports 4/1 - 4/4 ---------
Ingress Counters:
   Total Ingress Pkt Count:                  61402830423
   EnQue Pkt Count:                          61402825288
   DeQue Pkt Count:                          61402692118
   TotalQue Discard Pkt Count:               5096
   Oldest Discard Pkt Count:                 0
Egress Counters:
   EnQue Pkt Count:                          95406035820
   Discard Pkt Count:                        0

--------- Ports 4/5 - 4/8 ---------
Ingress Counters:
   Total Ingress Pkt Count:                  64207166485
   EnQue Pkt Count:                          64207161341
   DeQue Pkt Count:                          64207029336
   TotalQue Discard Pkt Count:               5087
   Oldest Discard Pkt Count:                 0
Egress Counters:
   EnQue Pkt Count:                          35018656764
   Discard Pkt Count:                        0
```

Syntax: show tm statistics [slot <*slot-number*>**]**

The **slot** <*slot-number*> option specifies the slot number of the port group that you want to display traffic manager statistics for.

Displaying traffic manager statistics for the 4x10G module

The following command displays traffic manager statistics for the 4x10G module identified by its slot number.

```
NetIron#show tm statistics slot 1
--------- Ports 1/1 - 1/2 ---------
Ingress Counters:
   Total Ingress Pkt Count:                  37145922200
   EnQue Pkt Count:                          37145922200
   EnQue Byte Count:                         5943609079168
   DeQue Pkt Count:                          37145922200
   DeQue Byte Count:                         5943609079168
   TotalQue Discard Pkt Count:               0
   TotalQue Discard Byte Count:              0
   Oldest Discard Pkt Count:                 0
   Oldest Discard Byte Count:                0
Egress Counters:
   EnQue Pkt Count:                          83890318963
   EnQue Byte Count:                         13422682341696
   Discard Pkt Count:                        218
   Discard Byte Count:                       34752

--------- Ports 1/3 - 1/4 ---------
Ingress Counters:
   Total Ingress Pkt Count:                  141547098478
   EnQue Pkt Count:                          141547098478
   EnQue Byte Count:                         22647526064544
   DeQue Pkt Count:                          141547098478
   DeQue Byte Count:                         22647526064544
   TotalQue Discard Pkt Count:               0
   TotalQue Discard Byte Count:              0
   Oldest Discard Pkt Count:                 0
   Oldest Discard Byte Count:                0
Egress Counters:
   EnQue Pkt Count:                          216769846687
   EnQue Byte Count:                         11606527206560
   Discard Pkt Count:                        0
   Discard Byte Count:                       0
```

Syntax: show tm statistics [slot <*slot-number*>**]**

The **slot** <*slot-number*> option specifies the slot number of the port group that you want to display traffic manager statistics for.

Displaying traffic manager statistics for the 24x1G module

The following command displays traffic manager statistics for the 24x1G module.

```
NetIron#show tm statistics all-counters 0
Ingress Counters:
   LBP Pkt Count:                          0
   QDP EnQue Pkt Count:                    0
   QDP EnQue Byte Count:                   0
   QDP DeQue Pkt Count:                    0
   QDP DeQue Byte Count:                   0
   QDP Head Delete Pkt Count:              0
   QDP Head Delete Byte Count:             0
   QDP Tail Delete Pkt Count:              0
   QDP Tail Delete Byte Count:             0
   Flow Status Message Count:              0
   Transmit Data Cell Count:               0
   TDM_A Pkt Count:                        0
   TDM_B Pkt Count:                        0

Programmable Ingress Counters:
[Queue Select: 8000, Queue Mask 0x0007]
   QDP EnQue Pkt Count:                    0
   QDP EnQue Byte Count:                   0
   QDP DeQue Pkt Count:                    0
   QDP DeQue Byte Count:                   0
   QDP Head Delete Pkt Count:              0
   QDP Head Delete Byte Count:             0
   QDP Tail Delete Pkt Count:              0
   QDP Tail Delete Byte Count:             0
   Flow Status Message Count:              0

Egress Counters:
   EGQ EnQue Pkt Count:                    0
   EGQ EnQue Byte Count:                   0
   EGQ Discard Pkt Count:                  0
   EGQ Discard Byte Count:                 0
   EGQ Segment Error Count:                0
   EGQ Fragment Error Count:               0
   Port63 Error Pkt Count:                 0
   Pkt Header Error Pkt Count:             0
   Pkt Lost Due to Buffer Full Pkt Count:  0
   Reassem Err Discard Pkt Count:          0
   Reassem Err Discard Fragment(32B) Count:  0
   TDM_A Lost Pkt Count:                   0
   TDM_B Lost Pkt Count:                   0

Programmable Egress Counters:
[Port Id for Enque: 0 (Disable), Port Id for Discard: 0 (Disable)]
   EGQ EnQue Pkt Count:                    0
   EGQ EnQue Byte Count:                   0
   EGQ Discard Pkt Count:                  0
   EGQ Discard Byte Count:                 0
```

Syntax: show tm statistics all-counters <*dev_id*>

The <*dev_id*> variable specifies the device id that you want to display traffic manager statistics for.

Clearing traffic manager statistics

You can clear traffic manager statistics selectively for a specified port group, selectively for an interface module, or for an entire PowerConnect router as shown in the following.

```
NetIron# clear tm statistics ethernet slot 4
```

Syntax: clear tm statistics [ethernet <*slot/port*> **| slot** <*slot-number*> **]**

Executing the **clear tm statistics** command without any options clears all traffic manager statistics on the router.

The **ethernet** <*slot/port*> option specifies a port group that you want to clear traffic manager statistics from.

The **slot** <*slot-number*> option specifies an interface module that you want to clear traffic manager statistics from.

New network processor counters displayed for packets to and from traffic manager

Output from the **show interface** command has been enhanced to provide the following traffic manager related information:

- Number of packets received at the network processor (NP)
- Number of packets sent from the NP to the traffic manager (TM)
- Number of Ingress packets dropped at the NP
- Number of packets transmitted from the NP
- Number of packets received by the NP from the TM

The following is an example of the new output from the show interface command with the changed output highlighted in **bold**.

```
NetIron(config)# show interface ethernet 3/3
GigabitEthernet3/3 is up, line protocol is up
Hardware is GigabitEthernet, address is 0004.80a0.4052 (bia 0004.80a0.4052)
Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
Member of L2 VLAN ID 1, port is untagged, port state is Forwarding
STP configured to ON, Priority is level0, flow control enabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
MTU 1544 bytes, encapsulation ethernet
300 second input rate: 754303848 bits/sec, 1473249 packets/sec, 89.57% utilization
300 second output rate: 754304283 bits/sec, 1473250 packets/sec, 89.57%
utilization
1015230949 packets input, 64974783168 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 1015230949 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants
NP received 1039220106 packets, Sent to TM 1039220442 packets
NP Ingress dropped 0 packets
1015231660 packets output, 64974824768 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 1015231660 unicasts
0 output errors, 0 collisions
NP transmitted 1039221393 packets, Received from TM 1039221562 packets
```

QoS for NI-MLX-1Gx48-T modules

The NI-MLX-1Gx48-T module supports 48 1G port. In a fully loaded 32 slot chassis, there are only 8 queues supported on the TM port. The PowerConnect chassis supports 1008 ports with 8 queues per port. Beginning with this release, the PowerConnect configuration allows you configure more ports in the system by changing the TM port to use 4 queues instead of 8. The PowerConnect chassis supports 2016 ports using 4 queues per port.

Limitations on TM ports

The TM Port limitations are reached under the following situations.

1. When a new module type is configured.

2. When a new line card is inserted (no configured type).

3. When the user tries to configure the **max-tm-queue** parameter from 4 to 8.

The relationship between max TM queues and max TM ports are supported in the system as follows:

TABLE 69 Maximum TM queues and TM ports

Max TM queue per port	Max TM port
8	1008
4	2016

Configuring priority queues from 8 to 4

The **system-init max-tm-queues** command allows you to configure the maximum number of queues in TM to 4. To configure priority queues from 8 to 4, enter the following command.

```
NetIron(config)# system-init max-tm-queues 4
```

Syntax: [no] system-init max-tm-queues <*num*>

The <*num*> value specifies changing the number of queues to 4.

NOTE
When configuring priority queues from 8 to 4, or vice versa, the system displays the following message: **Reload required. Please write memory and then reload or power cycle. Failure to reload could cause system instability or failure.**

The NP continues to map all inbound packets to 8 internal priorities. If the **system-init max-tm-queues** command is configured, the NP will right shift this priority number by one bit before sending the packet to TM. The TM will en-queue the packets based on the following table:

TABLE 70 Queue type

NP priority	TM queue
7,6	3
5,4	2

TABLE 70 Queue type (Continued)

NP priority	TM queue
3,2	1
1,0	0

QoS commands affected by priority queues

- Priority-based Rate Shaping
- Weighted Random Early Discard (WRED)
- Weighted-based Scheduling and Mixed Strict Priority
- CPU Copy Queue
- Traffic Manager Statistics

Priority-based rate shaping

If the user specifies a priority of 4-7 when the max-tm-queues parameter is configured using 4 queues, the **qos shaper priority** command is accepted, but a warning message is displayed.

The following example displays the **qos shaper priority** command configured with a priority 4 shaper.

```
NetIron(config-if-eth-16/1)# qos shaper priority 4 1000000
Warn: current max TM queues is 4-configuration of priority 4-7 will not have any
effect.
```

NOTE
If a priority 5 shaper is already configured, and the max-tm-queues parameter changes from 8 to 4 queues and is reloaded, then the priority 5 shaper configuration line is still displayed. The priority shaper 5 will not take effect.

Weighted Random Early Discard (WRED)

When WRED is enabled for a queue type of any forwarding queue, it will receive a warning message that is similar to when the **priority-based rate shaping** command is configured. Refer to "Priority-based rate shaping" on page 349 for more information.

The following example displays enabling WRED for the forwarding queues with a queue type of 6.

```
NetIron(config)#qos queue-type 6 wred enable
Warn: current max TM queues is 4-configuration of queue-type 4-7 will not have any
effect.
```

NOTE
If the **system-init max-tm-queues 4** command is configured, the user is able to configure similar WRED parameters, such as Average weight, Max Instantaneous queue size, Drop Precedence, etc. for all priorities. The default values of all WRED parameters (refer to Table 70 on page 348) is only effective when queue-type 0-3 is used.

Weighted-based scheduling and mixed strict priority

When the **max-tm-queues** parameter is configured with 8 or 4 queues, the **qos scheduler weighted** command and **qos scheduler mixed** command will still take the same number of weight values, but the unnecessary priority values are ignored.

The following example displays when the **qos scheduler weighted** command is configured using 4 queues.

```
NetIron(config-ethe-1/1)#qos scheduler weighted 7 6 5 4 3 2 11
Current max TM queues is 4 - weights “7”, “6”, “5”, “4” for priority 7-4 will not
have any effect.
```

The following example displays when the **qos scheduler mixed** command is configured using 4 queues.

```
NetIron(config-ethe-1/1)#qos scheduler mixed 4 3 2 1 1
Current max TM queues is 4 - weights “4”, “3”, “2” for queues 4-2 will not have
any effect.
```

The following table displays how traffic scheduling for Strict Priority-based Scheduling and Weighted-based Scheduling is configured differently between 8 and 4 queues:

TABLE 71 Strict v.s. weighted queues

	8 queues (current)	4 queues (new)
Strict	7,6,5	3,2
Weighted	4,3,2,1,0	1,0

Error messages for CPU copy queue and traffic manager statistics

The following error messages are displayed for CPU copy queue and traffic manager statistics when the incorrect queues are configured.

CPU copy queue

When **system-init max-tm-queues 4** command is configured, the **rl-cpu-copy** command displays a warning message when the user specifies a priority 4-7.

Example

```
NetIron(config)# rl-cpu-copy priority 4 1000000
Warn: current max TM queues is 4-configuration of priority 4-7 will not have any
effect.
```

Traffic manager statistics

When **system-init max-tm-queues 4** command is configured, the **show tm-voq-stat** command will only take a priority 0-3. An error message is displayed when an invalid priority range is enabled.

Example

```
NetIron#show tm-voq-stat src_port e 9/1 dst_port e 2/3 5
Error: priority range 0 to 3.
```

NOTE

The **show tm-voq-stat** command will print statistics for 4 queues, instead of 8. The output from the TM Q statistics is available only if the src card type is a 48x1GC module, or a POS module.

Configuring Traffic Policing for the NetIron MLX 10

Traffic policing on the PowerConnect

The PowerConnect Router provides line-rate traffic policing in hardware on inbound ports and outbound ports.

You can configure a PowerConnect Router to use one of the following modes of traffic policing policies:

- **Port-based** – Limits the rate on an individual physical port to a specified rate. Only one inbound and one outbound port-based traffic policing policy can be applied to a port. (Refer to "Configuring port-based traffic policing For inbound and outbound ports" on page 357.) These policies can be applied to inbound and outbound traffic.
- **Port-and-priority-based** – Limits the rate on an individual hardware forwarding queue on an individual physical port. Only one port-and-priority-based traffic policing policy can be specified per priority queue for a port. (Refer to "Configuring a port and priority-based traffic policing policy for inbound and outbound ports" on page 358.) These policies can be applied to inbound and outbound traffic.
- **VLAN-based** – Untagged packets as well as tagged packets can be rate-limited. Only one rate can be specified for each VLAN. (Refer to "Configuring a VLAN-based traffic policing policy" on page 359.) Up to 990 VLAN-based policies can be configured for a port under normal conditions or 3960 policies if priority-based traffic policing is disabled as described in "Configuring for no priority-based traffic policing" on page 362. These policies can be applied to inbound and outbound traffic.
- **VLAN group based** – Limits the traffic for a group of VLANs. Members of a VLAN group share the specified bandwidth defined in the traffic policing policy that has been applied to that group. (Refer to "Configuring a VLAN group-based traffic policing policy" on page 359.) Up to 990 VLAN Group-based policies can be configured for a port under normal conditions or 3960 policies if priority-based traffic policing is disabled as described in "Configuring for no priority-based traffic policing" on page 362. These policies can only be applied to inbound traffic.

  NOTE
  If a VLAN based policing is configured on a port for a particular VLAN, the policing will be applicable to all ports on that Network Processor that belong to that VLAN".

- **Port-and-ACL-based** – Limits the rate of IP traffic on an individual physical port that matches the permit conditions in IP Access Control Lists (ACLs). Layer 2 ACL-based traffic policing is supported. You can use standard or extended IP ACLs. Standard IP ACLs match traffic based on source IP address information. Extended ACLs match traffic based on source and destination IP address and IP protocol information. Extended ACLs for TCP and UDP also match on source and destination TCP or UDP addresses. and protocol information. These policies can be applied to inbound and outbound traffic. Up to 990 Port-and-ACL-based policies can be configured for a port under normal conditions or 3960 policies if priority-based traffic policing is disabled as described in "Configuring for no priority-based traffic policing" on page 362.

- **Rate Limiting for Copied-CPU-bound Traffic** – You can limit the rate of Copied-CPU-bound packets from applications such as sFlow, ACL logging, RPF logging, and source MAC address learning (with known destination address). Copied-CPU-bound packets are handled and queued separately from packets destined to the CPU such as protocol packets and using this feature they can be assigned to one of eight priority queues which has a rate limit assigned to it. The queue and rate are assigned by port and apply to all of the ports that are supported by the same packet processor. Table 72 describes the ports that are associated a packet processor.

Multi-Service IronWare supports applying traffic policing parameters directly to a port or creating a policy map to define a set of traffic policing parameters and then applying that policy map to one or more ports. In addition, the traffic policing parameters available from each of these options are different. The parameters used when applying traffic policing parameters directly to a port reflect the Multi-Service IronWare features that were available before this release. These parameters and the information required to use them are described in "Applying traffic policing parameters directly to a port" on page 354.

The parameters used when applying traffic policing through use of a policy map reflect the traffic policing features that have been added with this release. These parameters and the information required to use them are described in "Applying traffic policing parameters using a policy map" on page 355.

Applying traffic policing parameters directly to a port

When applying a traffic policing policy directly to a port, there are specific parameters that are applied to implement the policy that are different than those used when using a policy map. Using this method, a traffic policing policy specifies two parameters: average rate and maximum burst. These parameters are used to configure credits and credit totals.

Average rate

The *Average Rate* is the maximum number of bits a port is allowed to receive during a one-second interval. The rate of the traffic that matches the traffic policing policy will not exceed the average rate.

The Average Rate represents a percentage of an interface's line rate (bandwidth), expressed in bits per second (bps). It cannot be smaller than 8,144 bits per second (bps) and it cannot be larger than the port's line rate.

Average Rate must be entered in multiples of 8,144 bps. If you enter a number that is not a multiple of 8,144, the software adjusts the rate down to the lowest multiple of the number so that the calculation of credits does not result in a remainder of a partial Credit. For example, if you enter 10,000 bps, the value will be adjusted to 8,144 bps. The adjusted rate is sometimes called the *adjusted average rate*.

Maximum burst

Maximum burst provides a higher than average rate to traffic that meet the rate limiting criteria. Traffic will be allowed to pass through the port for a short period of time. The unused bandwidth can be accumulated up to a maximum of "maximum burst" value.

Credits and credit total

Each rate limiting policy is assigned a class. A *class* uses the average rate and maximum burst in the rate limit policy to calculate credits and credit totals.

Credit size is measured in bytes. A credit is a forwarding allowance for a traffic policed port, and is the smallest number of bytes that can be allowed during a rate limiting interval. Minimum credit size can be 1 byte.

During a rate limiting interval, a port can send or receive only as many bytes as the port has Credits for. For example, if an inbound rate limiting policy results in a port receiving two credits per rate limiting interval, the port can send or receive a maximum of 2 bytes of data during that interval.

In each interval, the number of bytes equal to the credit size is added to the running total of the class. The running total of a class represents the number of bytes that can be allowed to pass through without being subject to rate limiting.

The second parameter is the maximum *credit total*, which is also measured in bytes. The maximum credit total is based on the maximum burst value and is also measured in bytes.

The running total can never exceed the maximum credit total. When packets arrive at the port, a class is assigned to the packet based on the traffic policing policies. If the running total of the class is less than the size of the packet, then the packet is dropped. Otherwise, the size of the packet is subtracted from the running total and the packet is forwarded. If there is no traffic that matches traffic policing criteria, then the running total can grow up to the maximum credit total.

Applying traffic policing parameters using a policy map

When using the traffic policing policies available from previous versions, the policy parameters are provided explicitly for each port during port configuration. In this version, the policies must be defined using a policy map. The policy map configuration ties a policy name to a set of traffic policing policies. The policy name is then applied to the port or ports that you want to rate limit using the defined policy. This allows you to set a policy in a single location the affects multiple ports and to make changes to that policy. Configuration of a policy map is described in "Configuring a policy map" on page 356.

Within the policy map configuration, the parameters used to define traffic policing have been changed. When configuring traffic policing within a policy map, these new parameters apply. With this release, traffic policing policy determines the rate of inbound or outbound traffic (in bits per second or bps) that is allowed per port. This traffic is initially traffic policed by a Committed Information Rate (CIR) bucket. Traffic that is not accommodated in the CIR bucket is then subject to the Excess Information Rate (EIR) bucket.

The CIR bucket

The CIR rate limiting bucket is defined by two separate parameters: the CIR rate, and the Committed Burst Size (CBS) rate. The CIR rate is the maximum number of bits a port is allowed to receive or send during a one-second interval. The rate of the traffic that matches the traffic policing policy can not exceed the CIR rate. The CIR rate represents a portion of an interface's line rate (bandwidth), expressed in bits per second (bps) and it cannot be larger than the port's line rate. CIR-defined traffic that does not use the CIR rate available to it accumulates credits that it can use later in circumstances where it temporarily exceeds the CIR rate.

When traffic exceeds the bandwidth that has been reserved for it by the CIR rate defined in its policy, it becomes subject to the CBS rate. The CBS rate provides a rate higher than the CIR rate to traffic that exceeded its CIR rate. The bandwidth in the CBS rate is accumulated during periods of time when traffic that has been defined by a policy does not use the full CIR rate available to it. Traffic is allowed to pass through the port for a short period of time at the CBS rate.

When inbound or outbound traffic exceeds the bandwidth available for the defined CIR and CBS rates, it is either dropped, or made subject to the conditions set in it EIR bucket.

The EIR bucket

The EIR bucket provides an option for traffic that has exceeded the conditions set by policy for the CIR bucket. In the EIR bucket, there are two parameters that define the traffic that is available: the Excess Information Rate (EIR) and the Excess Burst Size (EBS) rate. The EIR and EBS operate exactly like the CIR and CBS except that they only act upon traffic that has been passed to the EIR bucket because it could not be accommodated by the CIR bucket. Like the CIR, the EIR provides an initial bandwidth allocation to accommodate inbound and outbound traffic. If the bandwidth provided by the EIR is insufficient to accommodate the excess traffic, the defined EBS rate provides for burst traffic. Like the CBS, the bandwidth available for burst traffic from the EBS is subject to the amount of bandwidth that is accumulated during periods of time when traffic that has been allocated by the EIR policy is not used.

In addition, to providing additional bandwidth for traffic that exceeds that available for the CIR bucket, traffic rate limited by the EIR bucket can have its excess priority and excess dscp values changed. Using this option, priority parameters are set following the EBS value that change the priority of traffic that is being rate limited using the EIR bucket.

Configuration considerations

- Only one type of traffic policing policy can be applied on a physical port. For example, you cannot apply port-and-ACL-based and port-based traffic policing policies on the same port.
- When a VLAN-based traffic policing policy is applied to a port, all the ports controlled by the same packet processor are rate limited for that VLAN. You cannot apply a VLAN-based traffic policing policy on another port of the same packet processor for the same VLAN ID.
- The Multi-Service IronWare software supports VLAN-based traffic policing that can limit tagged and untagged packets that match the VLAN ID specified in the policy. Untagged packets are not subject to traffic policing.
- The maximum burst in a traffic policing policy cannot be less than the average rate and cannot be more than the port's line rate.
- Control packets are not subject to traffic policing.
- Source MAC address with Virtual Leased Line (VLL) endpoints are not subject to traffic policing.

Configuring traffic policing on PowerConnect devices

The following sections show examples of how to configure each traffic policing policy type.

Configuring a policy map

To configure a policy map, enter a command such as the following.

```
NetIron(config)# policy-map map1 cir 1000000 cbs 2000000 eir 1000000 ebs 2000000
excess-dp 2 excess-dscp 37
```

The command configures the traffic policing policy map map1 to limit CIR rate to 1000000 the CBS rate to 2000000, the EIR rate to 1000000 and the EBS to 2000000. In addition, traffic that exceeds the bandwidth available in the CIR bucket will have its packets drop precedence set to 2 and its DSCP set to 37. This command only creates a policy, it must be applied to one or more ports to be operational.

Syntax: [no] **policy-map** <*map-name*> **cir** <*cir-rate*> **cbs** <*cbs-rate*> **[eir** <*eir-rate*> **ebs** <*ebs-rate*> **excess-priority** <*priority-num*> **[excess-dscp** <*dscp-num*>**]] | [eir** <*eir-rate*> **ebs** <*ebs-rate*> **excess-dp** <*dp-val*> **[excess-dscp** <*dscp-num*>**]]**

The map-name variable is the name you will use to reference the policy map in traffic policing command. It can be a character string up to 64 characters long.

The cir parameter defines the value of the Committed Information Rate (CIR) as the rate defined in the <*cir-rate*> variable. Acceptable values are: 0 - 10000000000 bps in increments of 8,144 bps.

The cbs parameter defines the value of the Committed Burst Size (CBS) as the rate defined in the <*cbs-rate*> variable. Acceptable values are: 1250 - 1250000000 bytes in increments of 1 byte.

The eir parameter defines the value of the Excess Information Rate (EIR) as the rate defined in the <*eir-rate*> variable. Acceptable values are: 0 - 10000000000 bps in increments of 8,144 bps.

The **ebs** parameter defines the value of the Excess Burst Size (EBS) as the rate defined in the <*ebs-rate*> variable. Acceptable values are: 1250 - 1250000000 bytes in increments of 1 byte.

The **excess-priority** parameter specifies that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket will have its packets priority queue set to the value set in the <*priority-num*> variable. Acceptable values for the <*priority-num*> are 0-7.

The **excess-dp** parameter specifies the WRED drop precedence for traffic whose bandwidth requirements exceed what is available in the CIR bucket and is sent to the EIR bucket. Acceptable values for the <*dp-value*> are 0-3. Packets with a value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped.

The **excess-dscp** parameter specifies that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket will have its packets DSCP priority set to the value set in the <*dscp-num*> variable. Acceptable values for the <*dscp-num*> are 0-63. When this parameter is used together with the **excess-dp** parameter, the value set for bits 2:1 (zero-based) in the **excess-dscp** parameter must be equal to the value set for **excess-dp**.

Configuring port-based traffic policing For inbound and outbound ports

Port-based traffic policing limits the rate on an individual inbound or outbound physical port to a specified rate.

To configure port-based traffic policing policy for outbound ports, enter commands such as the following at the interface level.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-1/1)# rate-limit out 500000000 750000000
```

The commands configure a traffic policing policy for outbound traffic on port 1/1. The policy limits the average rate of all outbound traffic to 500 Mbps with a maximum burst size of 750 Mbps.

To configure port based traffic policing policy through a policy map, enter a command such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-1/1)# rate-limit input policy-map map1
```

The commands configure a traffic policing policy for inbound traffic on port 1/1. The policy references the policy map map1 for rate limiting policy parameters.

The complete syntax for configuring a port-based traffic policing policy is:

Syntax: [no] rate-limit [in | out] [<*average-rate*> <*maximum-burst*> **| policy-map** <*map-name*>**]**

The **input** parameter applies the policy to traffic on inbound ports.

The **output** parameter applies the policy to traffic on outbound ports.

Only one inbound and one outbound port-based traffic policing policy can be applied to a port.

The <*average-rate*> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 8,144 bps. Refer to the section "Average rate" on page 354 for more details. This command is only used when configuring traffic policing directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

The <*maximum-burst*> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum burst" on page 354 for more details. This command is only used when configuring traffic policing directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

The **policy-map** parameter specifies the policy map named in the <*policy-map*> variable to be used to provide parameters for rate limiting the port and VLAN specified. This command is only used when configuring traffic policing to a port using a policy map as described in "Applying traffic policing parameters using a policy map" on page 355.

Configuring a port and priority-based traffic policing policy for inbound and outbound ports

To configure port based traffic policing policy directly, enter a command such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-1/1)# rate-limit input priority 2 500000000 750000000
```

The commands configure a traffic policing policy for inbound traffic on port 1/1. The policy limits the average rate of all inbound traffic to 500 Mbps with a maximum burst size of 750 Mbps for packets with their priority queue set to 2.

Syntax: **[no] rate-limit [input | output] priority** <*queue-num*> [<*average-rate*> <*maximum-burst*> | **policy-map** <*map-name*>]

The **input** parameter applies the policy to traffic on inbound ports.

The **output** parameter applies the policy to traffic on outbound ports.

Only one port-based traffic policing policy can be applied to a port.

The **priority** parameter specifies an 802.1p value in the <*queue-num*> variable that is used to identify packets that will be rate limited by this command.

The <*average-rate*> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the nearest multiple of 8,144 bps. Refer to the section "Average rate" on page 354 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

The <*maximum-burst*> parameter specifies the extra Mbits above the average-rate that traffic can have. Refer to the section "Maximum burst" on page 354 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

The **policy-map** parameter specifies the policy map named in the <*policy-map*> variable to be used to provide parameters for rate limiting the port. This command is only used when configuring rate limiting to a port using a policy map as described in "Applying traffic policing parameters using a policy map" on page 355.

Configuring a VLAN-based traffic policing policy

To configure a port-and-VLAN based traffic policing policy, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-1/1)# rate-limit input vlan 10 500000000 750000000

NetIron(config)# interface ethernet 1/2
NetIron(config-if-1/2)# rate-limit output vlan 20 policy-map map1
```

These commands configure two traffic policing policies that limit the average rate of all inbound traffic on port 1/1 with VLAN tag 10 and all outbound traffic on port 1/2 VLAN tag 20. The first policy limits packets with VLAN tag 10 to an average rate of 500 Mbps with a maximum burst size of 750 Mbits on port 1/1. The second policy limits packets with VLAN tag 20 to values defined in policy map map1. Tagged packets belonging to VLANs other than 10 and 20 and untagged packets are not subject to traffic policing on these ports.

Syntax: **[no] rate-limit [input | output] [priority** <*queue-num*>**] vlan-id** <*vlan-num*> **[**<*average-rate*> <*maximum-burst*> **| policy-map** <*map-name*> **]**

The **input** parameter applies the policy to traffic on inbound ports.

The **output** parameter applies the policy to traffic on outbound ports.

The **priority** parameter specifies an 802.1p value in the <*queue-num*> variable that is used to identify packets that will be rate limited by this command. This parameter is optional.

The vlan-id <*vlan-number*> parameter species the VLAN ID to which the policy applies. You can specify up to 990 priority or 3960 non-priority VLAN-based traffic policing policies on a port.

The <*average-rate*> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the nearest multiple of 8,144 bps. Refer to the section "Average rate" on page 354 for more details. This command is only used when configuring traffic policing directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

The <*maximum-burst*> parameter specifies the extra Mbits above the average-rate that traffic can have. Refer to the section "Maximum burst" on page 354 for more details. This command is only used when configuring traffic policing directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

The **policy-map** parameter specifies the policy map named in the <*policy-map*> variable to be used to provide parameters for traffic policing the port and VLAN specified. This command is only used when configuring traffic policing to a port using a policy map as described in "Applying traffic policing parameters using a policy map" on page 355.

Configuring a VLAN group-based traffic policing policy

A traffic policing policy can be applied to a VLAN group. VLANs that are members of a VLAN group share the specified bandwidth defined in the traffic policing policy applied to that group.

To configure a traffic policing policy for a VLAN group, perform the following tasks.

1. Define the VLANs that you want to place in a traffic policing VLAN group.
2. Define a rate limiting VLAN group. This VLAN group is specific to the traffic policing feature. Enter commands such as the following.

```
NetIron(config)# rl-vlan-group 10
NetIron(config-vlan-rate-group)# vlan 3 5 to 7 10
```

The commands assign VLANs 3, 5,6, 7, and 10 to traffic policing VLAN group 10.

Syntax: [no] rl-vlan-group <*vlan-group-number*>

Syntax: [no] vlan <*vlan-number*> **[to <vlan-number]**

The **rl-vlan-group** command takes you to the VLAN group traffic policing level. Enter the ID of the VLAN group that you want to create or update by entering a value for <*vlan-group-number*>.

Use the vlan command to assign or remove VLANs to the rate limiting VLAN group. You can enter the individual VLAN IDs or a range of VLAN IDs.

3. Create a policy for the VLAN group and apply it to the interface you want. Enter commands such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-1/1)# rate-limit input group 10 500000000 750000000
```

These commands configure a traffic policing policy that limits the average rate of all inbound traffic on port 1/1 from vlan group VlanGroupA. This policy limits packets from VlanGroupA to an average rate of 500 Mbps with a maximum burst size of 750 Mbits on port 1/1. VLAN Group based traffic policing is only available for inbound ports.

Syntax: [no] rate-limit input group <*vlan-group-id*> **[priority** <*queue-num*>**]** **[**<*average-rate*> <*maximum-burst*> **| policy-map** <*map-name*> **]**

The **input** parameter applies the policy to traffic on inbound ports.

The **priority** parameter specifies an 802.1p value in the <*queue-num*> variable that is used to identify packets that will be traffic policed by this command. This parameter is optional.

The <*vlan-group-id*> parameter species the VLAN GroupID to which the policy applies.

The <*average-rate*> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the nearest multiple of 8,144 bps. Refer to the section "Average rate" on page 354 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

The <*maximum-burst*> parameter specifies the extra Mbits above the average-rate that traffic can have. Refer to the section "Maximum burst" on page 354 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

The **policy-map** parameter specifies the policy map named in the <*policy-map*> variable to be used to provide parameters for rate limiting the VLAN specified. This command is only used when configuring traffic policing to a port using a policy map as described in "Applying traffic policing parameters using a policy map" on page 355.

Configuring a port and ACL-based rate limiting

You can use standard or extended IP ACLs for port-and-ACL-based rate limiting:

- Standard IP ACLs match traffic based on source IP address information.
- Extended ACLs match traffic based on source and destination IP addresses and IP protocol information. Extended ACLs for TCP and UDP protocol must also match on source and destination IP addresses and TCP or UDP protocol information.
- You can bind multiple rate limiting policies to a single port. However, once a matching ACL clause is found for a packet, the device does not evaluate subsequent clauses in that rate limiting ACL and subsequent rate limiting ACLs.

- You can apply an ACL ID to a port-and-ACL-based traffic policing policy even before you define the ACL. The traffic policing policy does not take effect until the ACL is defined.
- It is not necessary to remove an ACL from a port-and-ACL-based rate limiting policy before deleting the ACL.
- Layer 2 ACL rate limiting is supported.

Port-and-ACL-based traffic policing is supported for traffic on inbound and outbound ports. To configure port-and-ACL-based traffic policing policies, enter commands such as the following.

```
NetIron(config)#access-list 50 permit host 1.1.1.2
NetIron(config)#access-list 50 deny host 1.1.1.3
NetIron(config)#access-list 60 permit host 2.2.2.3
NetIron(config-if-1/1)# rate-limit input access-group 50 priority q1 500000000
750000000
NetIron(config-if-1/1)# rate-limit input access-group 60 100000000 200000000
```

These commands first configure access-list groups that contain the ACLs that will be used in the traffic policing policy. Use the **permit** condition for traffic that will be traffic policed. Traffic that match the **deny** condition are not subject to traffic policing.

Next, the commands configure two traffic policing policies on port 1/1. The policies limit the average rate of all inbound IP traffic that match the permit rules of ACLs 50 and 60. The first policy limits the rate of all permitted IP traffic with a priority queue value of q1 from host 1.1.1.2 to an average rate of 500 Mbps with a maximum burst size of 750 Mbits. Rate of all traffic from host 1.1.1.3 is not subject to rate limiting since it is denied by ACL 50; it is merely forwarded on the port.

The second policy limits the rate of all IP traffic from host 2.2.2.3 to an average rate of 100 Mbps with a maximum burst size of 200 Mbits.

All IP traffic that does not match ACLs 50 and 60 are not subject to traffic policing.

Syntax: [no] **rate-limit** [**input** | **output**] [**vrf** <*vrf-name*>] **access-group** <*group-number*> [**priority** <*queue-num*>] [<*average-rate*> <*maximum-burst*> | **policy-map** <*map-name*>]

The **input** parameter applies the policy to traffic on inbound ports.

The **output** parameter applies the policy to traffic on outbound ports.

The **VRF** parameter specifies that the access-group will only apply to traffic within the VRF whose name is specified in the <*vrf-name*> variable. This feature is only supported on inbound traffic with Layer-3 ACLs.

The **access-group**, group-number> parameter specifies the group number to which the ACLs used in the policy belong.

NOTE
An ACL must exist in the configuration before it can take effect in a traffic policing policy.

The **priority** parameter specifies a priority queue value in the <*queue-num*> variable that is used to identify packets that will be traffic policed by this command. The possible values for this parameter are: q0, q1, q2, or q3. Multiple queues can be specified. This parameter is optional.

The <*average-rate*> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the nearest multiple of 8,144 bps. Refer to the section "Average rate" on page 354 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354

The <*maximum-burst*> parameter specifies the extra Mbits above the average-rate that traffic can have. Refer to the section "Maximum burst" on page 354 for more details. This command is only used when configuring traffic policing directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354

The **policy-map** parameter specifies the policy map named in the <*policy-map*> variable to be used to provide parameters for traffic policing the VLAN specified. This command is only used when configuring traffic policing to a port using a policy map as described in "Applying traffic policing parameters using a policy map" on page 355.

Using ACLs for filtering in addition to rate limiting

When you use the ACL-based mode, the permit and deny conditions in an ACL you use in a rate limiting policy work as follows:

- **Permit** – The traffic is rate limited according to the other parameters in the rate limiting policy.
- **Deny** – The traffic is forwarded instead of dropped, by default.

You can configure the device to drop traffic that is denied by the ACL instead of forwarding the traffic, on an individual port basis.

NOTE
Once you configure an ACL-based rate limiting policy on a port, you cannot configure a regular (traffic filtering) ACL on the same port. To filter traffic, you must enable the strict ACL option.

To configure the device to drop traffic that is denied by a rate limiting ACL, enter the following command at the configuration level for the port.

```
NetIron(config-if-1/1)# rate-limit strict-acl
```

Syntax: [no] rate-limit strict-acl

Configuring for no priority-based traffic policing

By default, up to 990 different traffic policing policies can be applied to a single 10 GB Ethernet port. This combined with the 4 priorities utilizes 3960 rate limiting classes. You can configure a system-wide policy so that up to 3960 individual traffic policing policies can be applied to a single 10 GB Ethernet port.

To configure a PowerConnect Router to not allow priority-based traffic policing, enter commands such as the following at the interface level.

```
NetIron(config)# qos-policy
NetIron(qos-policy)# no rate-limit internal-priority-based
```

Syntax: [no] rate-limit internal-priority-based

If this command is implemented, the number of different rate limiting policies that can be applied to a single port is increased from 990 to 3960.

Configuring rate limiting for Copied-CPU-bound traffic

A new feature was added that allows you to limit the rate of Copied-CPU-bound packets from applications such as sFlow, ACL logging, RPF logging, and source MAC address learning (with known destination address). This feature can be configured as described in this section

The following command assigns a rate limit of 200,000,000 bps and a priority queue of 0 to copied-CPU-bound incoming traffic on PPCR 1 though its assignment on port 3/2.

```
NetIron(config)# rl-cpu-copy 0 200000000 ethernet 3/2
```

Syntax: rl-cpu-copy <*priority-number*> <*limit-rate*> **ethernet** <*slot/port*> **[to ethernet** <*slot/port*> **]**

The <*priority-number*> variable specifies the CPU-bound traffic priority queue to apply the rate limiting. This can be a value from 0 to 7.

The <*limit-rate*> variable specifies the limiting rate for the specified CPU-bound traffic priority queue. Acceptable values are from 1 to 300000000 bps. The default rate for all is 300,000,000 bps.

The <*slot/port*> variable specifies the port that you want to apply copied-CPU-bound rate limiting to. You can apply the command to a range of ports using the **to ethernet** <*slot/port*> option. When you assign a port, the command applies to all ports that are associated with the same packet processor (PPCR). Table 72 describes the ports that are associated a packet processor.

TABLE 72 Ports per packet processor

Interface module	Ports Per Packet Processor (PPCR)	
	PPCR1	PPCR2
4 X 10 Gbps	1 - 2	3 - 4
20 X 1 Gbps	1 - 20	

You can display the **rl-cpu-copy** configuration displayed previously using the following command.

```
NetIron# show rl-cpu-copy
Rate shaping configuration on CPU Copy priority queues
priority 0 200000000 ethernet 3/1 to 3/20
```

Notice that although the command was only executed for port 3/2, it applies to all the ports attached to the same PPCR. In this case ports 3/1 to 3/20.

Syntax: show rl-cpu-copy

Displaying rate limiting policies

Use one of the following commands to view the rate limiting policies that have been configured:

- **show rate limit counters** – Displays accounting information for rate limit usage.
- **show rate limit group** – Displays the VLANs that are in the specified group.
- **show rate limit** – Displays rate limiting policies implemented per interface.
- **show policy map** – Displays rate limiting policies implemented in the configured policy maps.

You can configure an PowerConnect router to exclude the 20-byte per-packet Ethernet overhead from Traffic Policing byte accounting. This can be done by configuring the **vlan-counter exclude-overhead** command.

Displaying accounting information for rate limit usage

To display accounting information for rate limit usage, enter the following command.

```
NetIron# show rate-limit counters
```

Syntax: show rate-limit counters [interface slot/port]

The **interface slot/port** option allows you to get accounting information for a specified interface only.

Output such as the following will display.

```
NetIron# show rate-limit counters
interface e 2/1
 rate-limit input access-group 400 999993616 1000000000
  Fwd:        0                       Drop:  0 bytes
  Re-mark:    0                       Total: 0 bytes
```

This display shows the following information.

TABLE 73 Rate limit counters parameters

This field...	Displays...
Interface	The interface that rate limit accounting information is being displayed for.
rate-limit input	A rate limit configuration that defines rate limit policy for inbound traffic on the defined interface.
Fwd	The traffic in bytes that has been forwarded from this interface as a result of this rate limit policy since the router was started up or the counter has been reset.
Drop	The traffic in bytes that has been dropped from this interface as a result of the defined rate limit policy since the router was started up or the counter has been reset.
Re-mark	The number of packets whose priority have been remarked as a result of exceeding the bandwidth available in the CIR bucket for this rate limit policy.
Total	The total traffic in bytes that has been carried on this interface for the defined rate limit policy since the router was started up or the counter has been reset.

Monitoring rate limit usage by SNMP

Accounting information for rate limit usage can also be monitored by SNMP. The agAclAccntTable supports the following objects, which map to the rate limit usage counters.

agAclAccntRaclDropCnt: drop counters

agAclAccntRaclFwdCnt: fwd counters

agAclAccntRaclRemarkCnt: re-mark counters

agAclAccntRaclTotalCnt: total counters

For detailed information, please refer to the “Filtering Traffic” chapter of the *Fabric OS MIB Reference*.

Among other applications, this accounting feature allows per-port VLAN statistics in the inbound or outbound direction to be extracted by means of SNMP. This can be achieved by adding ACL filters for the monitored VLAN on the appropriate port. This accounting feature works for all modules of the NetIron MLX platform. For modules that do not support extended VLAN statistics, this feature provides a means of extracting per-port VLAN statistics.

Resetting the rate limit counters

You can reset all of the rate limit counters using the following command.

```
NetIron# clear rate-limit counters
```

Syntax: clear rate-limit counters [interface]

The **interface** variable specifies an interface that you want to clear the rate limit counters for. If you do not specify an interface, all rate limit counters on the router will be reset.

Displaying information about rate limit VLAN groups

To display information about rate limit VLAN groups, enter the following command.

```
NetIron# show rate-limit group
```

Syntax: show rate-limit group

Output such as the following will display

```
rl-vlan-group 1
 vlan 10 to 15
```

This display shows the following information.

TABLE 74 Rate limit VLAN group parameters

This field...	Displays...
rl-vlan-group	The VLAN group whose contents are displayed.
vlan	VLANs contained in the VLAN group specified.

Displaying rate limit policies per interface

To display information about rate limit policies that are configured per interface, enter the following command.

```
NetIron# show rate-limit
```

Syntax: show rate-limit

Output such as the following will display.

```
NetIron(config-if-e10000-1/1)#show rate-limit

interface e 1/1
 rate-limit input 959904 2000000
 rate-limit output 2986368 2000000
```

This display shows the following information.

TABLE 75 Rate limit interface parameters

This field...	Displays...
rate-limit input	The average-rate and maximum burst rate configured for inbound traffic on the specified interface.
rate-limit output	The average-rate and maximum burst rate configured for outbound traffic on the specified interface.

Displaying rate limit policies configured in policy maps

To display information about rate limit policy maps, enter the following command.

```
NetIron# show policy-map
```

Syntax: show policy-map [map-name]

The <*map-name*> variable limits the display of policy map configuration information to the map specified. If this variable is not used, configuration information will be displayed for all policy maps configured on the router.

Output such as the following will display.

```
NetIron(config-policymap pmap1)#show policy-map

policy-map pmap1
  cir 106656       bps cbs 24000        bytes
  eir 53328        bps ebs 20000        bytes
  excess-priority 2 excess-dscp 43

policy-map pmap2
  cir 106656       bps cbs 24000        bytes
  eir 53328        bps ebs 30000        bytes
  excess-priority 1 excess-dscp 30
```

This display shows the following information.

TABLE 76 Rate limit policy map parameters

This field...	Displays...
policy-map	The name of the policy map whose configuration is being displayed
cir	The value of the Committed Information Rate (CIR) configured for this policy map. Possible values are: 1 - 10000000000 bps.
cbs	The value of the Committed Burst Size (CBS) configured for this policy map. Possible values are: 1250 - 1250000000 bytes.
eir	The value of the Excess Information Rate (EIR) configured for this policy map. Possible values are: 1 - 10000000000 bps.
ebs	The value of the Excess Burst Size (EBS) configured for this policy map. Possible values are: 1250 - 1250000000 bytes.
excess-priority	The priority queue that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket is set to. Possible values are 0-3.
excess-dscp	The priority queue that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket is set to. Possible values are 0-63.

Layer 2 ACL-based rate limiting

Layer 2 ACL-based rate limiting enables devices to limit the rate of incoming traffic in hardware, without CPU intervention. Rate limiting in hardware enables the device to manage bandwidth at line-rate speed.

In general, Layer 2 ACL-based rate limiting works along the same lines as hardware-based rate limiting feature. All the rules and regulations that apply to hardware-based rate limiting also apply to this feature.

Configuration rules and notes

- You can apply Layer 2 ACL-based rate limiting on a physical port. You cannot apply it to a virtual interface or a LAG port.
- You cannot use IPv4 ACL-based filtering and IPv4 Layer 2 ACL-based rate limiting on the same port. You can, however, configure one port on the device to use IP ACLs and another port on the same device to use Layer 2 ACL-based rate limiting.
- You cannot use IPv4 ACL-based rate limiting and IPv4 Layer 2 ACL-based rate limiting on the same port. You can, however, configure one port on the device to use IPv4 ACL-base rate limitng and another port on the same device to use Layer 2 ACL-based rate limiting.
- You can bind multiple rate limiting policies to a single port. However, once a matching ACL clause is found for a packet, the device does not evaluate subsequent clauses in that rate limiting ACL and subsequent rate limiting ACLs.
- Only number ACLs support rate limiting
- Layer 2 rate limiting ACLs will function with vlan-cpu-protection, broadcast and multicast limiting features. If incoming traffic matches an inbound Layer 2 rate limiting ACL, it is first rate-limited based on the policy. If packets are not dropped due to rate limiting, they are forwarded either to the CPU or flooded in the VLAN according to the vlan-cpu-protection feature.
- The broadcast and multicast packet limiting feature limits packets in the CPU, while the Layer 2 ACL RL is a network processing (NP) RL feature. Packets are first subjected to the Layer 2 ACL RL at the NP. Once packets are forwarded to CPU, the broadcast and multicast limiting feature begins functioning and packets may be dropped in the CPU if the rate exceeds the limit.

Editing a Layer 2 ACL Table

You can make changes to the Layer 2 ACL table definitions without unbinding and rebinding the rate limit policy. For example, you can add a new clause to the ACL table, delete a clause from the table, or delete the ACL table that is used by a rate limit policy.

Define rate limiting parameters

To define rate limiting parameters, enter commands such as the following:

```
NetIron(config)#policy-map map1
NetIron(config-policymap map1)#cir 1000000 cbs 2000000 eir 1000000 ebs 2000000
excess-dp 2
```

Binding Layer 2 ACL-based rate limiting policy to a port

To bind an Layer 2 ACL based rate-limiting policy on a specific port, enter commands such as the following:

```
NetIron(config-policymap map1)#int eth 14/1
NetIron(config-if-e10000-14/1)# rate-limit input access-group 400 policy-map map1
```

Syntax: [no}rate-limit [input|output] access-group <num> policy-map <map-name>

Specifying rate limiting parameters without a policy map

To specify rate-limiting without using a policy map, enter a command such as the following:

```
NetIron(config-if-e10000-14/1)# rate-limit input access-group 400 49999998416
75000000000
```

Syntax: [no] rate-limit [input|output] access-group <acl-id> <average-rate> <maximum-burst>

The <acl-id> for Layer 2 ACLs can range from 400 to 499.

The <average-rate> is the maximum number of bits the policy allows during one second.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum burst" on page 354 for more details. This command is only used when configuring traffic policing directly to a port as described in "Applying traffic policing parameters directly to a port" on page 354.

Display accounting

To display access list accounting, enter a command such as the following.

```
PowerConnect#show access-list accounting eth 14/1 in rate-limit
Collecting L2 ACL accounting for 400 on port 14/1  ...  Completed successfully.
RL ACL Accounting Information:
Inbound: ACL 400
    0:   permit 0000.0000.0021 ffff.ffff.ffff any any etype any
       Hit count: (1 sec)                    0   (1 min)                    0
                  (5 min)                    0   (accum)                    0
```

Rate limiting protocol traffic using Layer 2 inbound ACLs

Using interface level Layer 2 inbound ACLs, you can rate limit the following types of protocol traffic by explicitly configuring a filter to match the traffic:

- STP/RSTP/BPDU
- MRP
- VSRP
- LACP
- GARP
- UDLP

To rate-limit all such control traffic enter commands such as the following:

```
PowerConnect(config)#access-list 402 permit any 0180.c200.0000 ffff.ffff.ffff any
etype any
PowerConnect(config)#access-list 402 permit any 0304.8000.0000 ffff.ffff.ffff any
etype any
PowerConnect(config)#access-list 402 permit any 0304.8000.0100 ffff.ffff.ff00 any
etype any
PowerConnect(config)#access-list 402 permit any 0180.c200.0002 ffff.ffff.ffff any
etype any
PowerConnect(config)#access-list 402 permit any 0180.c200.0020 ffff.ffff.fff0 any
etype any
```

```
PowerConnect(config)#access-list 402 permit any 00e0.5200.0000 ffff.ffff.ffff any
etype any
PowerConnect(config)#access-list 402 deny any any any etype any
```

Table 77 lists the protocols and their corresponding filters.

TABLE 77 Filters for protocols

Protocol	Filter
STP/RSTP/BPDU	access-list 402 permit any 0180.c200.0000 ffff.ffff.ffff any etype any
MRP	access-list 402 permit any 0304.8000.0000 ffff.ffff.ffff any etype any
VSRP	access-list 402 permit any 0304.8000.0100 ffff.ffff.ff00 any etype any
LACP	access-list 402 permit any 0180.c200.0002 ffff.ffff.ffff any etype any
GARP	access-list 402 permit any 0180.c200.0020 ffff.ffff.fff0 any etype any
UDLP	access-list 402 permit any 00e0.5200.0000 ffff.ffff.ffff any etype any

NOTE
The filters must have the specific destination MAC address as shown above in the configuration. You can filter all protocols as shown in the previous configuration example above, or only specific protocols.

Example of Layer 2 ACL to rate limit broadcast traffic

To define an ACL that rate limits broadcast traffic and forwards all other traffic without rate limiting, enter commands such the following:

```
NetIron(config)#access-list 411 permit any ffff.ffff.ffff ffff.ffff.ffff
NetIron(config)#access-list 411 deny any any
```

To bind an ACL that rate limits broadcast traffic and forwards all other traffic without rate limiting, enter commands such the following

```
NetIron(config)#int eth 14/1
NetIron(config-if-e10000-14/1)#rate-limit in access-gr 411 8144 100
```

Rate limiting ARP packets

You can limit the rate of ARP traffic that requires CPU processing on PowerConnect devices, such as ARP request traffic, and ARP response addressed to the router. The feature is set globally and applies to all ARP traffic received at the device. With this feature you can apply a defined policy map to all ARP traffic bound for the CPU.

When the **vlan-cpu-protection** command is configured, ARP request packets are switched within a VLAN by the hardware and thus cannot be rate-limited by the **ip rate limit arp policy-map** command. To limit the rate of ARP packets that are forwarded by hardware, use interface-level, layer-2 inbound ACLs with the "etype arp" option.

Configuring rate limiting of ARP packets

To rate limit ARP packets bound for the CPU using a policy map named “limitarp”, enter the following command.

```
NetIron(config)# ip rate-limit arp policy-map limitarp
```

Syntax: [no] ip rate-limit arp policy-map <map-name>

The <map-name> variable is the name of the policy map to be used to provide parameters for rate limiting CPU-bound ARP packets. If the policy map specified has not been defined, the rate limit values are initialized to the line rate values.

Displaying statistics for ARP rate limiting

You can display ARP Rate Limiting Statistics using the following command.

```
NetIron# show rate-limit arp
 Fwd:        1865392                Drop:  867731400 bytes
 Re-mark:    1864800                Total: 871461592 bytes
```

Syntax: show rate-limit arp

This display shows the following information.

TABLE 78 Rate limit ARP display parameters

Parameter	Description
Fwd	The ARP traffic in bytes that has been sent to the CPU as a result of the ARP rate limit policy since the router was started up or the counter was reset.
Drop	The ARP traffic in bytes that has been dropped as a result of the ARP rate limit policy since the router was started up or the counter was reset.
Re-mark	The ARP traffic in bytes whose priority have been remarked as a result of exceed the bandwidth available in the CIR bucket for the ARP rate limit policy since the router was started up or the counter was reset.
Total	The total ARP traffic in bytes that has been subjected to the ARP rate limit policy since the router was started up or the counter was reset.

Clearing Statistics for ARP Rate Limiting

You can clear ARP Rate Limiting Statistics using the following command:

```
NetIron# clear rate-limit arp
```

Syntax: clear rate-limit arp

Configuring Spanning Tree Protocol 11

Overview

The following STP features are supported by NetIron MLX Series devices.

- IEEE 802.1D Spanning Tree Protocol (STP)
- STP per VLAN
- STP Fast Forwarding
- STP Enable or Disable per Port or VLAN
- Root Guard
- BPDU Guard
- IEEE Single Spanning Tree (SSTP)
- SuperSpan
- PVST or PVST+ Compatibility
- 802.1s Multiple Spanning Tree Protocol
- STP support under an ESI with support for B-VLANs, S-VLANs and C-VLANs

A user can configure ESIs in the process of configuring Provider Bridging and Provider Backbone Bridging. By default, a device has a "default ESI" configured in which VLANs 1- 4090 exist. This chapter refers to configuration and use of Spanning Tree Protocols under the default ESI.

IEEE 802.1D Spanning Tree Protocol (STP)

The PowerConnect router supports Spanning Tree Protocol (STP) as described in the IEEE 802.10-1998 specification. STP eliminates Layer 2 loops in networks, by selectively blocking some ports and allowing other ports to forward traffic, based on configurable bridge and port parameters. STP also ensures that the least cost path is taken when multiple paths exist between ports or VLANs. If the selected path fails, STP searches for and then establishes an alternate path to prevent or limit retransmission of data.

Enabling or disabling STP

STP is disabled by default on the PowerConnect. Thus, new VLANs you configure on the PowerConnect have STP disabled by default. Table 79 lists the default STP states for the PowerConnect

TABLE 79 Default STP states

Device type	Default STP type	Default STP state	Default STP state of new VLANs
PowerConnect	Dell's multiple instances of spanning tree	Disabled	Disabled

By default, each VLAN on a PowerConnect runs a separate spanning tree instance. Each PowerConnect has one VLAN (VLAN 1) by default that contains all of its ports. However, if you configure additional port-based VLANs on a PowerConnect, then each of those VLANs on which STP is enabled and VLAN 1 all run separate spanning trees.

You can enable or disable STP on the following levels:

- **Globally** – Affects all VLANs on the PowerConnect.
- **Individual VLAN** – Affects all ports within the specified VLAN. When you enable or disable STP within a VLAN, the setting overrides the global setting. Thus, you can enable STP for the ports within a VLAN even when STP is globally disabled, or disable the ports within a port-based VLAN when STP is globally enabled.
- **Individual port** – Affects only the individual port. However, if you change the STP state of the primary port in a LAG group, the change affects all ports in the LAG group.

Enabling or disabling STP globally

Use the following methods to enable or disable STP on the PowerConnect on which you have not configured VLANs.

NOTE
When you configure a VLAN, the VLAN inherits the global STP settings. However, once you begin to define a VLAN, you can no longer configure standard STP parameters globally using the CLI. From that point on, you can configure STP only within individual VLANs.

NOTE
Reloading the PowerConnect switch with global STP enabled can display error on boot-up (error - no more stp instances available) if the number of vlans in the configuration are more than configured **system-max** for STP instances. The error message has no effect on the functionality.

When configuring spanning- tree at the global CLI level, the following message will prompt you to enter "y" for yes or "n" for no to change the spanning-tree behavior at the global level:

```
NetIron(config)#spanning-tree
This will change the spanning-tree behavior at the global level.
Are you sure? (enter 'y' or 'n'): y
```

Enter 'y' to change the spanning-tree behavior. Enter 'n' to make no change to the spanning-tree configuration at the global level.

This command enables a separate spanning tree in each VLAN, including the default VLAN.

Syntax: [no] **spanning-tree**

Enabling or disabling STP on a VLAN

Use the following procedure to disable or enable STP on a PowerConnect on which you have configured a VLAN. Changing the STP state in a VLAN affects only that VLAN.

To enable STP for all ports in a port-based VLAN, enter commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# spanning-tree

that there is no effect on the functionality due to this error message
```

Syntax: [no] spanning-tree

Enabling or disabling STP on a port

Use the following procedure to disable or enable STP on an individual port.

NOTE
If you change the STP state of the primary port in a LAG group, the change affects all ports in the LAG group.

To enable STP on an individual port, enter commands such as the following.

```
NetIron(config)# interface 1/1
NetIron(config-if-e1000-1/1)# spanning-tree
```

Syntax: [no] spanning-tree

Default STP bridge and port parameters

Table 80 lists the default STP bridge parameters. The bridge parameters affect the entire spanning tree. If you are using MSTP, the parameters affect the VLAN. If you are using SSTP, the parameters affect all VLANs that are members of the single spanning tree.

NOTE
STP information is specific to a VLAN, and the MultiService IronWare software uses the CONTROL VLAN to get the STP information for the all MIBs under dot1dStp and dot1DStpPortTable. Due to the limitation of the MIBs, information of per STP implementation of every specific VLAN is not displayed.

TABLE 80 Default STP bridge parameters

Parameter	Description	Default and valid values
Forward Delay	The period of time a bridge will wait (the listen and learn period) before beginning to forward data packets.	15 seconds Possible values: 4 – 30 seconds
Maximum Age	The interval a bridge will wait for a hello packet from the root bridge before initiating a topology change.	20 seconds Possible values: 6 – 40 seconds
Hello Time	The interval of time between each configuration BPDU sent by the root bridge.	2 seconds Possible values: 1 – 10 seconds
Priority	A parameter used to identify the root bridge in a spanning tree (instance of STP). The bridge with the lowest value has the highest priority and is the root. A higher numerical value means a lower priority; thus, the highest priority is 0.	32768 Possible values: 0 – 65535

NOTE
If you plan to change STP bridge timers, it is recommended that you stay within the following ranges, from section 8.10.2 of the IEEE specification:

– 2 * (forward_delay -1) >= max_age
– max_age >= 2 * (hello_time +1)

Table 81 lists the default STP port parameters. The port parameters affect individual ports and are separately configurable on each port.

TABLE 81 Default STP port parameters

Parameter	Description	Default and valid values
Priority	The preference that STP gives this port relative to other ports for forwarding traffic out of the spanning tree. A higher numerical value means a lower priority; thus, the highest priority is 8.	128 Possible values: 8 – 252, configurable in increments of 4
Path Cost	The cost of using the port to reach the root bridge. When selecting among multiple links to the root bridge, STP chooses the link with the lowest path cost and blocks the other paths. Each port type has its own default STP path cost.	10 Mbps – 100 100 Mbps – 19 Gigabit – 4 10 Gigabit – 2 Possible values are 1– 65535

Changing STP bridge parameters

To change a PowerConnect's STP bridge priority to the highest value, so as to make the PowerConnect the root bridge, enter the following command.

```
NetIron(config)# vlan 20
NetIron(config-vlan-20)# spanning-tree priority 0
```

To make this change in the default VLAN, enter the following commands.

```
NetIron(config)# vlan 1
NetIron(config-vlan-1)# spanning-tree priority 0
```

Syntax: **[no] spanning-tree [forward-delay** <*value*>**] | [hello-time** <*value*>**] | [max-age** <*value*>**] | [priority** <*value*>**]**

You can specify some or all of the parameters on the same command line. For information on parameters, possible values and defaults, refer to Table 80 on page 373.

NOTE
The **hello-time** <*value*> parameter applies only when the device or VLAN is the root bridge for its spanning tree.

Changing STP port parameters

To change the path and priority costs for a port, enter commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# spanning-tree ethernet 1/5 path-cost 15 priority 64
```

Syntax: **[no] spanning-tree ethernet** <*slot*>/<*portnum*> **path-cost** <*value*> **| priority** <*value*> **| disable | enable**

The **ethernet** <*slot*>/<*portnum*> parameter specifies the interface.

For descriptions of path cost and priority, their default and possible values, refer to Table 81 on page 374. If you enter a priority value that is not divisible by four, the software rounds it to the nearest value.

The **disable | enable** parameter disables or re-enables STP on the port. The STP state change affects only this VLAN. The port's STP state in other VLANs is not changed.

Root Guard

In this release, a new security feature has been added that allows a port to run STP but not allow the connected device to become the Root. The Root Guard feature provides a way to enforce the root bridge placement in the network and allows STP to interoperate with user network bridges while still maintaining the bridged network topology that the administrator requires. Errors are triggered if any change from the root bridge placement is detected.

NOTE
The feature is also available for MSTP and RSTP.

When Root Guard is enabled on a port, it keeps the port in designated FORWARDING state. If the port receives a superior BPDU, which is a Root Guard violation, it sets the port into BLOCKING state and triggers a Syslog message and an SNMP trap. No further traffic will be forwarded on this port. This allows the bridge to prevent traffic from being forwarded on ports connected to rogue or misconfigured STP or RSTP bridges.

Root Guard should be configured on all ports where the root bridge should not appear. In this way, the core bridged network can be cut off from the user network by establishing a protective perimeter around it.

Once the port stops receiving superior BPDUs, Root Guard will automatically set the port back to a FORWARDING state after the timeout period has expired.

NOTE
Root Guard may prevent network connectivity if improperly configured. It needs to be configured on the perimeter of the network rather than the core. Also, Root Guard should be configured only on the primary port of a LAG. If a port configured with Root Guard is made a secondary port, the LAG deployment will be vetoed.

Enabling Root Guard

Root Guard is configured on a per interfaces basis. To enable Root Guard, enter a command such as the following.

```
NetIron(config)# interface ethernet 5/5
NetIron(config-if-e10000-5/5) spanning-tree root-protect
```

Syntax: [no] spanning-tree root-protect

Enter the **no** form of the command to disable Root Guard on the port.

Setting the Root Guard timeout period

To configure the Root Guard timeout period globally, enter a command such as the following.

```
NetIron(config)# spanning-tree root-protect timeout 120
```

Syntax: [no] spanning-tree root-protect timeout <*timeout in seconds*>

The **timeout in seconds** parameter allows you to set the timeout period. The timeout period may be configured to anything between 5 and 600 seconds. Default is 30 seconds.

Checking if Root Guard is configured

To determine if Root Guard is configured, enter the following command.

```
NetIron#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is enabled, STP BPDU Guard is disabled
```

Syntax: show interface ethernet <*slot*>/<*port*>

Displaying the Root Guard state

To display the Root Guard state, enter the **show spanning-tree root-protect** command.

```
NetIron#show spanning-tree root-protect
Port VLAN Current State

13/6 3 Consistent state
13/9 2 Inconsistent state (29 seconds left on timer)
```

Syntax: show spanning-tree root-protect

Reconfiguring the timeout period

The timeout period timer is activated whenever a port encounters a superior BPDU, which then results in a Root Guard violation. If the timeout period is reconfigured while a timer is in use, the timer on that port is set to the new timeout period, minus the time elapsed since the superior BPDU was received.

For example, the original timeout period on a device was configured for 60 seconds. The port encounters a superior BPDU and the timer starts. Issuing a **show span root-protect** CLI command displays the following information.

```
NetIron(config)#show span root-protect

Port    VLAN  Current State

1/4     1     Inconsistent state (56 seconds left on timer)
```

While the timer is in use, the timeout period is changed to 30 seconds through the issue of the following command.

```
NetIron(config)# spanning-tree root-protect timeout 30
```

The timer continues the countdown and minus the time that have already elapsed (about 10 seconds) since the superior BPDU was detected. Issuing a **show span root-protect** CLI command displays the following information.

```
NetIron(config)# show span root-protect

Port    VLAN  Current State

1/4     1     Inconsistent state (20 seconds left on timer)
```

Next, the timeout period is increased to 120 seconds.

```
NetIron(config)# spanning-tree root-protect timeout 120
```

Since the timer has not expired, it continues the countdown. The remaining time left is adjusted by the time that has already elapsed (about 18 seconds) since the superior BPDU was detected. Issuing a **show span root-protect** CLI command displays the following information.

```
NetIron(config)# show span root-protect

Port    VLAN  Current State

1/4     1     Inconsistent state (102 seconds left on timer)
```

Checking for Syslog messages

A Syslog message such as the following is generated after the Root Guard blocks a port.

```
Sep 9 18::39:27:I:STP: Root Guard Port 12/21, VLAN 10 inconsistent (Received
superior BPDU)
```

A Syslog message such as the following is generated after the Root Guard unblocks a port.

```
Sep 9 18::39:27:I:STP: Root Guard Port 12/21, VLAN 10 consistent (Timeout)
```

Checking for traps

The following SNMP traps are generated for Root Guard:

- snTrapStpRootGuardDetect is generated after the Root Guard blocks a port.
- snTrapStpRootGuardExpire is generated after a blocked port (due to Root Guard) goes back to a Forwarding state

Refer to the *IronWare MIB Reference* manual for details.

BPDU Guard

STP protection provides the ability to prohibit an end station from initiating or participating in an STP topology. The Bridge Protocol Data Units (BPDU) Guard is used to keep all active network topologies predictable.

NOTE
The feature is also available for MSTP and RSTP.

STP detects and eliminates logical loops in a redundant network by selectively blocking some data paths and allowing only some data paths to forward traffic.

In an STP environment, switches, end stations, and other Layer 2 devices use BPDUs to exchange information that STP will use to determine the best path for data flow. When a Layer 2 device is powered ON and connected to the network, or when a Layer 2 device goes down, it sends out an BPDU, triggering a topology change.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in a topology change. In this case, you can enable the BPDU Guard feature on the Dell port to which the end station is connected. The BPDU Guard feature disables the connected device's ability to initiate or participate in an topology change, by dropping all BPDUs received from the connected device.

As an extended security measure, the administrator can disable a port if a BPDU is received on a port where BPDU Guard is configured. A Syslog message and SNMP trap are triggered when the port is disabled.

You can re-enable the disabled port from the CLI; however, make sure the offending BPDUs have stopped before re-enabling the port. Otherwise, the port will be disabled again the moment a new BPDU is received.

NOTE
BPDU Guard should be configured only on the primary port of a LAG. If a port configured with BPDU guard is made a secondary port, the LAG deployment will be vetoed.

Enabling BPDU Guard

You can enable BPDU Guard on a per-port basis.

To prevent an end station from initiating or participating in topology changes, enter the following command at the interface level of the CLI.

```
NetIron(config) interface ethe 2/1
NetIron(config-if-e1000-2/1)# spanning-tree protect
```

Syntax: [no] spanning-tree protect

This command causes the port to drop BPDUs sent from the device on the other end of the link.

Enter the **no** form of the command to disable BPDU Guard on the port and remove the **spanning-tree protect do-disable** feature if they are configured.

Enabling BPDU Guard and disabling a port that receives BPDUs

You can enable BPDU Guard on a port and at the same time configure a port to be disabled when it receives a BPDU. Enter the following commands.

```
NetIron(config) interface ethe 2/1
NetIron(config-if-e1000-2/1)#spanning-tree protect do-disable
```

Syntax: [no] spanning-tree protect do-disable

If both **spanning-tree protect** and **spanning-tree protect do-disable** are configured on an interface, **spanning-tree protect do-disable** takes precedence. This means that when the port receives a BPDU, the port will drop the BPDU and disable the port.

If you issue a **no spanning-tree protect do-disable** command, the port will be re-enabled and will no longer be disabled when it receives a BPDU. The following message is displayed when you enter the **no spanning-tree protect do-disable** command.

```
This command removes only "spanning-tree protect do-disable". To remove
"spanning-tree protect", please issue a separate command "no spanning-tree
protect".
```

Re-Enabling a port disabled due to BPDU guard

A port disabled by the **spanning-tree protect do-disable** command can be enabled by the following commands:

- Entering the **no spanning-tree protect do-disable** command.

- Entering the **spanning-tree protect re-enable** command. Make sure the offending BPDUs have stopped before issuing this command; otherwise, the port will be disabled again once it receives a new BPDU.

```
NetIron(config)# interface ethernet 1/4
NetIron(config-if-e10000-1/4)#spanning-tree protect re-enable
```

Syntax: [no] spanning-tree protect re-enable

Issuing the **spanning-tree protect re-enable** command does not remove the **spanning-tree protect do-disable** configuration on the port. If a new BPDU is received on the port, the port will be disabled again. To prevent this from happening, you can do one of the following:

- Remove the **spanning-tree protect do-disable** configuration by issuing the **no spanning-tree protect do-disable** command, followed by the **spanning-tree protect re-enable** command to re-enable the port.
- Remove the source of the offending BPDUs from the network.

There is no **no** form of this command.

Displaying BPDU Guard configuration

To determine if BPDU Guard is configured on the device, enter the following command.

```
NetIron#show spanning-tree protect

  protect   Show STP BPDU Guard information
NetIronr# show span protect
Port    Disable Port on BPDU Rx       Current Port State
1/1     No                            down
1/2     Yes                           down
1/3     No                            up
1/4     Yes                           up
```

Syntax: show spanning-tree protect

The command shows the following information.

TABLE 82 CLI display of show spanning-tree bp

This field...	Displays...
Port	The port on which BPDU Guard is configured
Disable Port on BPDU Rx	Indicates if **spanning-tree protect do-disable** is configured on the port: • Yes - **spanning-tree protect do-disable** is configured on the port. The BPDU will be dropped and the port will be disabled when it receives a BPDU. • No - **spanning-tree protect do-disable** is not configured. The BPDU will be dropped but the port will not be disabled.
Current Port State	Indicates if the port is currently UP or DOWN.

Determining if BPDU Guard is enabled

The **show interface** command displays the state of a port.

If BPDU Guard is disabled or has not been configured, the output shows the following information.

```
NetIron#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
```

If BPDU Guard has been enabled using the **spanning-tree protect** command, the output shows the following.

```
NetIron#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is enabled
```

If BPDU Guard is enabled using the **spanning-tree protect do-disable** command, the output shows.

```
NetIron#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is enabled with port to be disabled
on BPDU receive
```

Syntax: show interface ethernet <*slot*>/<*port*>

Checking for Syslog messages

When the **spanning-tree protect do-disable** command is issued, the port becomes disabled and the following Syslog messages are generated.

```
Sep 9 18::39:27:I:STP: BPDU Guard port 1/4 disable
Sep 9 18::39:27:I:System: Interface ethernet 1/4, state down - disabled
```

When the **spanning-tree protect re-enable** is issued to re-enable a port, the following Syslog messages are generated.

```
Sep 9 18:43:21:I:STP: BPDU Guard re-enabled on ports ethe 1/4
Sep 9 18:43:23:I:System: Interface ethernet 1/4, state up
```

Checking for traps

The following SNMP traps are generated for BPDU Guard:

- snTrapStpBPDUGuardDetect is generated when a port is disabled because **spanning-tree protect do-disable** on a port and that port received a BPDU and disabled the port.
- snTrapSTPBPDUGuardExpire is generated when a port that has been disabled due to a BPDU Guard violation is re-enabled using the **spanning-tree protect re-enable** command.

Refer to the *IronWare MIB Reference* manual for details.

Displaying STP information

You can display the following STP information:

- All the global and interface STP settings
- Detailed STP information for each interface
- STP state information for a VLAN
- STP state information for an individual interface

Displaying STP information for an entire device

To display STP information, enter the following command at any level of the CLI.

```
NetIron# show spanning-tree vlan 10

VLAN 10 - STP instance 1
------------------------------------------------------------------
STP Bridge Parameters:

Bridge           Bridge Bridge Bridge Hold  LastTopology Topology
Identifier       MaxAge Hello  FwdDly Time  Change       Change
hex              sec    sec    sec    sec   sec          cnt
8000000480a04000 20     2      15     1     0            0

RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
8000000480a04000 0         8000000480a04000 Root  20  2   15

STP Port Parameters:

Port  Prio Path      State      Designat- Designated       Designated
Num   rity Cost                ed Cost   Root             Bridge
1/3   128  4         DISABLED   0         0000000000000000 0000000000000000
1/13  128  4         DISABLED   0         0000000000000000 0000000000000000
```

Syntax: **show spanning-tree** [**vlan** <*vlan-id*>] | [**pvst-mode**] | [<*num*>] | [**detail** [**vlan** <*vlan-id*> [**ethernet** <*slot/port*>] [| **begin**<*expression*> | **exclude**<*expression*> | **include**<*expression*>]

The **vlan** <*vlan-id*> parameter displays STP information for the specified port-based VLAN.

The **pvst-mode** parameter displays STP information for the PowerConnect's Per VLAN Spanning Tree (PVST+) compatibility configuration. Refer to "PVST or PVST+ compatibility" on page 396.

The <*num*> parameter displays only the entries after the number you specify. For example, on a PowerConnect with three port-based VLANs, if you enter 1, then information for the second and third VLANs is displayed, but information for the first VLAN is not displayed. Information is displayed according to VLAN number, in ascending order. The entry number is not the same as the VLAN number. For example, if you have port-based VLANs 1, 10, and 2024, then the command output has three STP entries. To display information for VLANs 10 and 2024 only, enter **show spanning-tree 1**.

The **detail** parameter and its additional optional parameters display detailed information for individual ports. Refer to "Displaying detailed STP information for each interface" on page 384.

The **show spanning-tree** command shows the following information.

TABLE 83 CLI display of STP information

This field...	Displays...
Global STP Parameters	
VLAN ID	The port-based VLAN that contains this spanning tree and the number of STP instance on the VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all STP information is for VLAN 1.
Bridge Parameters	

TABLE 83 CLI display of STP information (Continued)

This field...	Displays...
Bridge Identifier	The ID assigned by STP to this bridge for this spanning tree in hexadecimal. **NOTE:** If this address is the same as the Root ID, then this device or VLAN is the root bridge for its spanning tree.
Bridge MaxAge sec	The number of seconds this bridge waits for a hello message from the root bridge before deciding the root has become unavailable and performing a reconvergence.
Bridge Hello sec	The interval between each configuration BPDU sent by the bridge.
Bridge FwdDly sec	The number of seconds this bridge waits following a topology change and consequent reconvergence.
Hold Time sec	The minimum number of seconds that must elapse between transmissions of consecutive Configuration BPDUs on a port.
Last Topology Chang sec	The number of seconds since the last time a topology change occurred.
Topology Change cnt	The number of times the topology has changed since this device was reloaded.
Root Bridge Parameters	
Root Identifier	The ID assigned by STP to the root bridge for this spanning tree in hexadecimal.
Root Cost	The cumulative cost from this bridge to the root bridge. If this device is the root bridge, then the root cost is 0.
DesignatedBridge Identifier	The designated bridge to which the root port is connected. The designated bridge is the device that connects the network segment on the port to the root bridge.
Root Port	The port on this device that connects to the root bridge. If this device is the root bridge, then the value is “Root” instead of a port number.
Max Age sec	The number of seconds this root bridge waits for a hello message from the bridges before deciding a bridges has become unavailable and performing a reconvergence.
Hello sec	The interval between each configuration BPDU sent by the root bridge.
FwdDly sec	The number of seconds this root bridge waits following a topology change and consequent reconvergence.
Port STP Parameters	
Port Num	The port number.
Priority	The port’s STP priority. **NOTE:** If you configure this value, specify it in decimal format. Refer to “Changing STP port parameters” on page 374.
Path Cost	The port’s STP path cost.

TABLE 83 CLI display of STP information (Continued)

This field...	Displays...
State	The port's STP state. The state can be one of the following: • BLOCKING – STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED – The port is not participating in STP. This can occur when the port is disconnected or STP is disabled on the port. • FORWARDING – STP is allowing the port to send and receive frames. • LISTENING – STP is responding to a topology change and this port is listening for a BPDU from neighboring bridges in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING – The port has passed through the LISTENING state and will change to the BLOCKING or FORWARDING state, depending on the results of STP's reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
Design Cost	The cost to the root bridge as advertised by the designated bridge that is connected to this port. If the designated bridge is the root bridge itself, then the cost is 0. The identity of the designated bridge is shown in the Design Bridge field.
Designated Root	The root bridge as recognized on this port. The value is the same as the root bridge ID listed in the Root ID field.
Designated Bridge	The bridge as recognized on this port.

Displaying detailed STP information for each interface

To display the detailed STP information, enter the following command at any level of the CLI.

```
NetIron# show spanning-tree detail vlan 10
VLAN 10 - STP instance 1
------------------------------------------------------------------------
STP Bridge Parameters:

Bridge identifier - 0x8000000480a04000
Root bridge - 0x8000000480a04000
Control ports - ethernet 1/3 ethernet 1/13
Active global timers - None

STP Port Parameters:

Port 1/3 - DISABLED
Port 1/13 - DISABLED

VLAN 20 - STP instance 2
------------------------------------------------------------------------
STP Bridge Parameters:

Bridge identifier - 0x8000000480a04000
Root bridge - 0x8000000480a04000
Control ports - ethernet 1/3 ethernet 1/13
Active global timers - None

STP Port Parameters:

Port 1/3 - DISABLED
Port 1/13 - DISABLED
```

If a port is disabled, the only information shown by this command is “DISABLED”. If a port is enabled, this display shows the following information.

Syntax: show spanning-tree detail [vlan <*vlan-id*> **[ethernet** <*slot/port*>**]]**

The **vlan** <*vlan-id*> parameter specifies a VLAN.

The **ethernet** <*slot*>/<*portnum*> parameter specifies an individual port within the VLAN (if specified).

The <*num*> parameter specifies the number of VLANs you want the CLI to skip before displaying detailed STP information. For example, if the PowerConnect has six VLANs configured (VLAN IDs 1, 2, 3, 99, 128, and 256) and you enter the command **show span detail 4**, detailed STP information is displayed for VLANs 128 and 256 only.

NOTE
If the configuration includes VLAN groups, the **show span detail** command displays the master VLANs of each group but not the member VLANs within the groups. However, the command does indicate that the VLAN is a master VLAN. The **show span detail vlan** <*vlan-id*> command displays the information for the VLAN even if it is a member VLAN. To list all the member VLANs within a VLAN group, enter the **show vlan-group** [<*group-id*>] command.

The **show spanning-tree detail** command shows the following information for each VLAN participating in the spanning tree.

TABLE 84 CLI display of detailed STP information for ports

This field...	Displays...
VLAN ID	The VLAN that contains the listed ports and the number of STP instances on this VLAN. The STP type can be one of the following: • Proprietary multiple Spanning Tree • IEEE 802.1Q Single Spanning Tree (SSTP) **NOTE:** If STP is disabled on a VLAN, the command displays the following message instead: "Spanning-tree of port-vlan <*vlan-id*> is disabled."
STP Bridge Parameters:	
Bridge identifier	The STP identity of this device.
Root	The ID assigned by STP to the root bridge for this spanning tree.
Control ports	The ports in the VLAN.
Active global timers	The global STP timers that are currently active, and their current values. The following timers can be listed: • Hello – The interval between Hello packets. This timer applies only to the root bridge. • Topology Change (TC) – The amount of time during which the topology change flag in Hello packets will be marked, indicating a topology change. This timer applies only to the root bridge. • Topology Change Notification (TCN) – The interval between Topology Change Notification packets sent by a non-root bridge toward the root bridge. This timer applies only to non-root bridges.
STP Port Parameters:	
Port number and STP state	The internal port number and the port's STP state. The internal port number is one of the following: • The port's interface number, if the port is the designated port for the LAN. • The interface number of the designated port from the received BPDU, if the interface is not the designated port for the LAN. The state can be one of the following: • BLOCKING – STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED – The port is not participating in STP. This can occur when the port is disconnected or STP is administratively disabled on the port. • FORWARDING – STP is allowing the port to send and receive frames. • LISTENING – STP is responding to a topology change and this port is listening for a BPDU from neighboring bridges in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING – The port has passed through the LISTENING state and will change to the BLOCKING or FORWARDING state, depending on the results of STP's reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table. **NOTE:** If the state is DISABLED, no further STP information is displayed for the port.

IEEE Single Spanning Tree (SSTP)

By default, each port-based VLAN on the PowerConnect runs a separate spanning tree, which you can enable or disable on an individual VLAN basis.

Alternatively, you can configure the PowerConnect to run a single spanning tree across all of its ports and VLANs. The SSTP feature is especially useful for connecting a PowerConnect to third-party devices that run a single spanning tree in accordance with the 802.1q specification.

SSTP uses the same parameters, with the same value ranges and defaults, as the default STP supported on the PowerConnect. Refer to "Default STP bridge and port parameters" on page 373.

SSTP defaults

SSTP is disabled by default. When you enable the feature, all VLANs on which STP is enabled become members of a single spanning tree. All VLANs on which STP is disabled are excluded from the single spanning tree:

- To add a VLAN to the single spanning tree, enable STP on that VLAN.
- To remove a VLAN from the single spanning tree, disable STP on that VLAN.

When you enable SSTP, all the ports that are in port-based VLANs with STP enabled become members of a single spanning tree domain. Thus, the ports share a single BPDU broadcast domain. The PowerConnect places all the ports in a non-configurable VLAN, 4095, to implement the SSTP domain. However, this VLAN does not affect port membership in the port-based VLANs you have configured. Other broadcast traffic is still contained within the individual port-based VLANs. Therefore, you can use SSTP while still using your existing VLAN configurations without changing your network. In addition, SSTP does not affect 802.1q tagging. Tagged and untagged ports alike can be members of the single spanning tree domain.

NOTE
When SSTP is enabled, the BPDUs on tagged ports go out untagged.

If you disable SSTP, all VLANs that were members of the single spanning tree now do not run any form of spanning tree. Per VLAN STP can be enabled again using either global STP enable or the spanning-tree enable command under individual VLANs.

NOTE
If the PowerConnect has only one port-based VLAN (the default VLAN), then it is already running a single instance of STP. In this case, you do not need to enable SSTP. You need to enable SSTP only if the PowerConnect contains more than one port-based VLAN and you want all the ports to be in the same STP broadcast domain.

To configure the PowerConnect to run a single spanning tree, enter the following command at the global CONFIG level.

```
NetIron(config)# spanning-tree single
```

To change a global STP parameter, enter a command such as the following at the global CONFIG level.

```
NetIron(config) spanning-tree single priority 2
```

This command changes the STP priority for all ports to 2.

To change an STP parameter for a specific port, enter commands such as the following.

```
NetIron(config) spanning-tree single ethernet 1/1 priority 10
```

The commands shown above override the global setting for STP priority and set the priority to 10 for port 1/1.

Here is the syntax for the global STP parameters:

Syntax: **[no] spanning-tree single [forward-delay** <*value*>**]**
[hello-time <*value*>**] | [maximum-age** <*time*>**] | [priority** <*value*>**]**

Here is the syntax for the STP port parameters:

Syntax: **[no] spanning-tree single [ethernet** <*slot*>**/**<*portnum*> **path-cost** <*value*> **| priority** <*value*>**]**

For the parameter definitions and possible values, refer to "Default STP port parameters" on page 374.

NOTE
Both commands listed above are entered at the global CONFIG level.

Also, you can use the **rstp single** command to control the topology for VLANs. Refer to "Enabling or disabling RSTP on a single spanning tree" on page 443.

Displaying SSTP information

To verify that SSTP is in effect, enter the following commands at any level of the CLI.

```
NetIron(config)# show spanning-tree
VLAN 4095 - STP instance 0
------------------------------------------------------------------
STP Bridge Parameters:

Bridge           Bridge Bridge Bridge Hold  LastTopology Topology
Identifier       MaxAge Hello  FwdDly Time  Change       Change
hex              sec    sec    sec    sec   sec          cnt
8000000480a04000 20     2      15     1     0            0

RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
8000000480a04000 0         8000000480a04000 Root  20  2   15

STP Port Parameters:

Port  Prio Path      State      Designat- Designated       Designated
Num   rity Cost                ed Cost   Root             Bridge
1/3   128  4         DISABLED   0         0000000000000000 0000000000000000
1/13  128  4         DISABLED   0         0000000000000000 0000000000000000

SSTP members: 10 20 30 99 to 100
```

For information on the command syntax, refer to "Displaying STP information" on page 380.

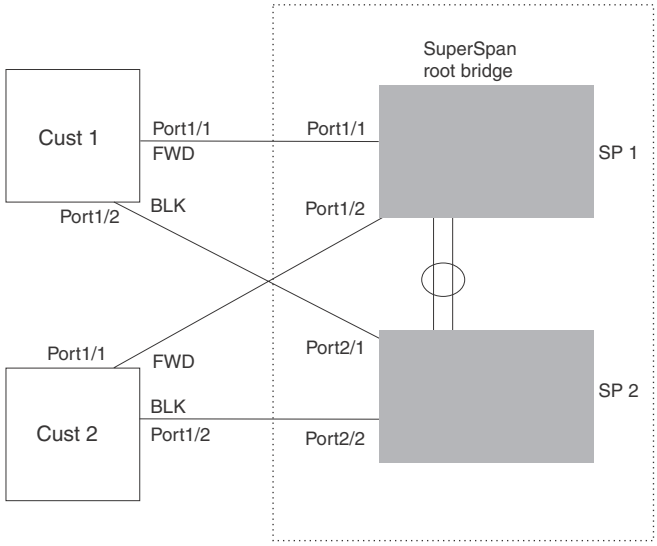
SuperSpan™

SuperSpan is a Dell STP enhancement that allows Service Providers (SPs) to use STP in both SP networks and customer networks. The SP devices are PowerConnect devices and are configured to tunnel each customers' STP BPDUs through the SP. From the customer's perspective, the SP network is a loop-free non-blocking device or network. The SP network behaves like a hub in the sense that the necessary blocking occurs in the customer network, not in the SP.

The interfaces that connect the SP to a customer's network are configured as SuperSpan boundary interfaces. Each SuperSpan boundary interface is configured with a customer ID, to uniquely identify the customer's network within SuperSpan.

Figure 20 shows an example SuperSpan implementation. In this example, an SP's network is connected to multiple customers. Each customer network is running its own instance of standard STP. The PowerConnect devices in the SP are running SuperSpan.

FIGURE 20 SuperSpan example



In this example, the SP network contains two devices that are running SuperSpan. The SP is connected to two customer networks. Each customer network is running its own instance of STP. SuperSpan prevents Layer 2 loops in the traffic flow with each customer while at the same time isolating each customer's traffic and spanning tree from the traffic and spanning trees of other customers. For example, the SP devices provide loop prevention for Customer 1 while ensuring that Customer 1's traffic is never forwarded to Customer 2. In this example, customer 1 has two interfaces to the SP network, ports 1/1 and 1/2 connected to SP 1. The SP network behaves like a non-blocking hub. BPDUs are tunneled through the network. To prevent a Layer 2 loop, customer 1's port 1/2 enters the blocking state.

Customer ID

SuperSpan uses a SuperSpan customer ID to uniquely identify and forward traffic for each customer. You assign the customer ID as part of the SuperSpan configuration of the PowerConnect devices in the SP. In Figure 20, the spanning trees of customer 1 and customer 2 do not interfere with one another because the SP network isolates each customer's spanning tree based on the SuperSpan customer IDs in the traffic.

BPDU forwarding

When the PowerConnect receives a customer's BPDU on a boundary interface, the PowerConnect changes the destination MAC address of the BPDU from the bridge group address (01-80-c2-00-00-00) as follows:

- The first byte (locally administered bit) is changed from 01 to 03, to indicate that the BPDU needs to be tunneled.
- The fourth and fifth bytes are changed to the customer STP ID specified on the boundary interface.

For example, if the customer's STP ID is 1, the destination MAC address of the customer's BPDUs is changed to the following: 03-80-c2-00-01-00.

Each PowerConnect that is configured for SuperSpan forwards the BPDU using the changed destination MAC address. At the other end of the tunnel, the PowerConnect connected to the customer's network changes the destination MAC address back to the bridge group address (01-80-c2-00-00-00).

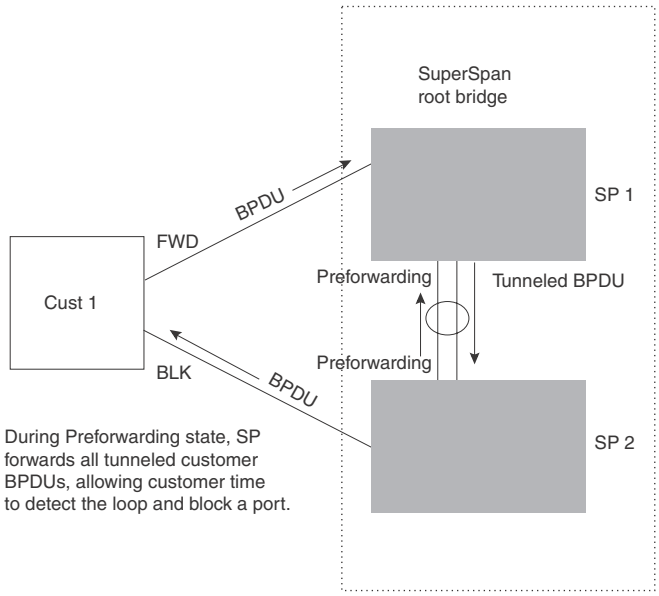
Preforwarding state

To ensure that the customer's network has time to converge at Layer 2 and prevent loops, the PowerConnect devices configured for SuperSpan use a special forwarding state, Preforwarding. The Preforwarding state occurs between the Learning and Forwarding states and by default lasts for five seconds. During the Preforwarding state, the PowerConnect forwards tunneled BPDUs from customers only and does not forward data traffic. This ensures that the customer's network will detect the Layer 2 loop and block a port. The SP network remains unblocked. After the Preforwarding state, the ports change to the Forwarding state and forward data traffic as well as BPDUs.

The default length of the Preforwarding state is five seconds. You can change the length of the Preforwarding state to a value from 3 – 30 seconds.

Figure 21 shows an example of how the Preforwarding state is used.

FIGURE 21 SuperSpan Preforwarding state



In this example, a customer has two links to the SP. Since the SP is running SuperSpan, the SP ports enter the Preforwarding state briefly to allow the customer ports connected to the SP to detect the Layer 2 loop and block one of the ports.

NOTE
If you add a new PowerConnect to a network that is already running SuperSpan, you must enable SuperSpan on the PowerConnect, at least on the VLANs that will be tunneling the customer traffic. Otherwise, the new PowerConnect does not use the Preforwarding state. This can cause the wrong ports to be blocked.

Combining single STP and multiple spanning trees

You can use SuperSpan in any of the following combinations:

- Customer and SP networks both use multiple spanning trees (a separate spanning tree in each VLAN).
- Customer uses multiple spanning trees but SP uses Single STP (all STP-enabled VLANs are in the same spanning tree).
- Customer uses Single STP but SP uses multiple spanning trees.
- Customer and SP networks both use Single STP.

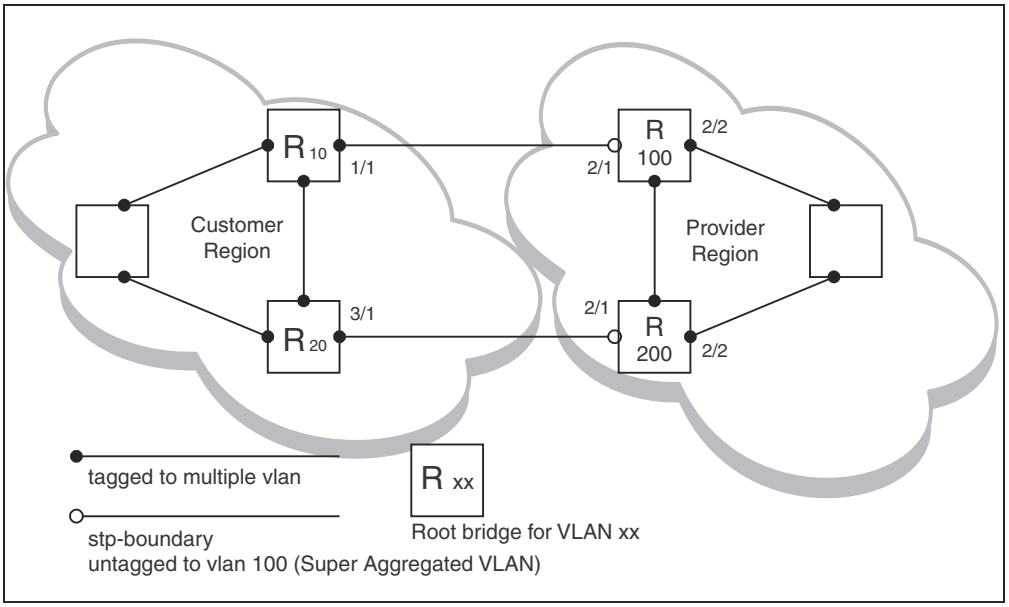
The following sections provide an example of each combination.

NOTE
All the combinations listed above are supported when the boundary ports joining the SP SuperSpan domain to the client spanning trees are untagged. For example, all these combinations are valid in super aggregated VLAN configurations. If the boundary ports are tagged, you cannot use Single STP in the client network in combination with multiple spanning trees in the SP SuperSpan domain.

Customer and SP use multiple spanning trees

Figure 22 shows an example of SuperSpan where both the customer network and the SP network use multiple spanning trees (a separate spanning tree in each port-based VLAN).

FIGURE 22 Customer and SP using multiple spanning trees



Both the customer and SP regions are running multiple spanning trees (one per port-based VLAN) in the Layer 2 switched network. The customer network contains VLANs 10 and 20 while the SP network contains VLANs 100 and 200. Customer traffic from VLAN 10 and VLAN 20 is aggregated by VLAN 100 in the SP since the boundary ports, 2/1 on R100 and R200, are untagged members of VLAN 100. By adjusting the bridge priority on VLANs 10 and 20, the customer can select a different root bridge for each spanning tree running in the customer network.

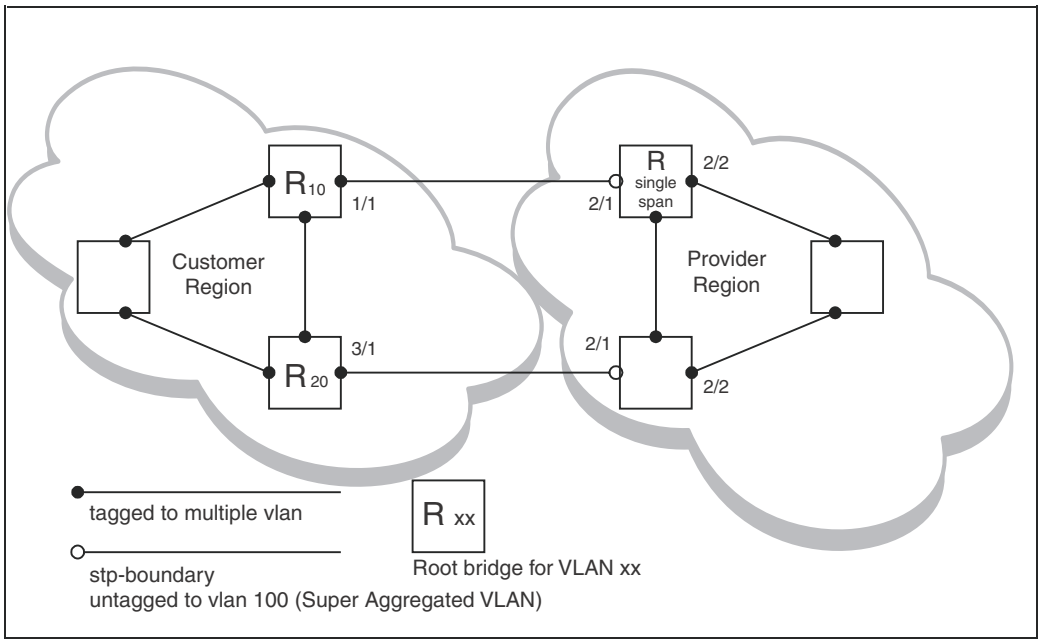
In the above example, STP in VLAN 10 will select R10 as the root bridge and make 1/1 on R10 forwarding while blocking port 3/1 on R20. The opposite occurs for STP in VLAN 20. As a result, both links connecting the customer and SP regions are fully utilized and serve as backup links at the same time, providing loop-free, non-blocking connectivity. In the SP network, multiple STP instances are running (one for VLAN 100 and one for VLAN 200) to ensure loop-free, non-blocking connectivity in each VLAN.

SuperSPAN boundaries are configured at port 2/1 of R100 and R200. Since the customer's traffic will be aggregated into VLAN 100 at the SP, the SP network appears to the customer to be a loop-free non-blocking hub to the customer network when port 2/2 on R200 is blocked by STP in VLAN 100.

Customer uses multiple spanning trees but SP uses single STP

Figure 23 shows an example of SuperSpan where the customer network uses multiple spanning trees while the SP network uses Single STP.

FIGURE 23 Customer using multiple spanning trees and SP using Single STP



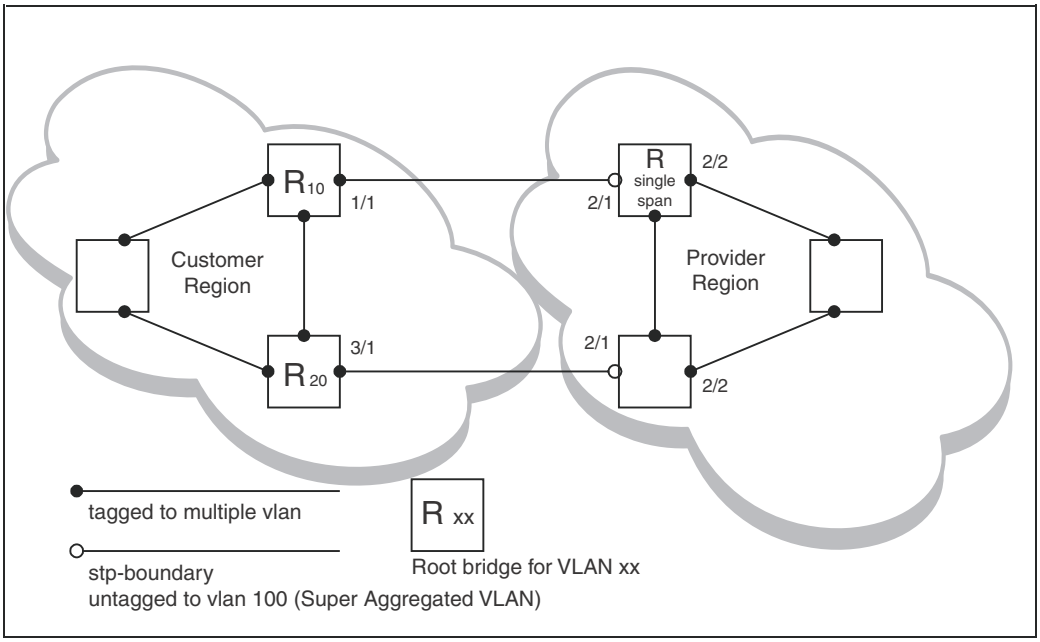
Customer traffic from different VLANs is maintained by different spanning trees, while the SP network is maintained by a single spanning tree. The SP can still use multiple VLANs at the core to separate traffic from different customers. However, all VLANs will have the same network topology because they are all calculated by the single spanning tree. The loop-free, non-blocking network acts like a hub for the customer network, with boundary ports 2/1 on each device being untagged members of VLAN 100.

Traffic from all VLANs in the customer network will be aggregated through VLAN 100 at the SP. This setup leaves the customer network's switching pattern virtually unchanged from the scenario in "Customer and SP use multiple spanning trees" on page 391, since the SP network still is perceived as a virtual hub, and maintenance of the hub's loop-free topology is transparent to the customer network.

Customer uses single STP but SP uses multiple spanning trees

Figure 24 shows an example of SuperSpan where the customer network uses Single STP while the SP uses multiple spanning trees.

FIGURE 24 Customer using Single STP and SP using multiple spanning trees

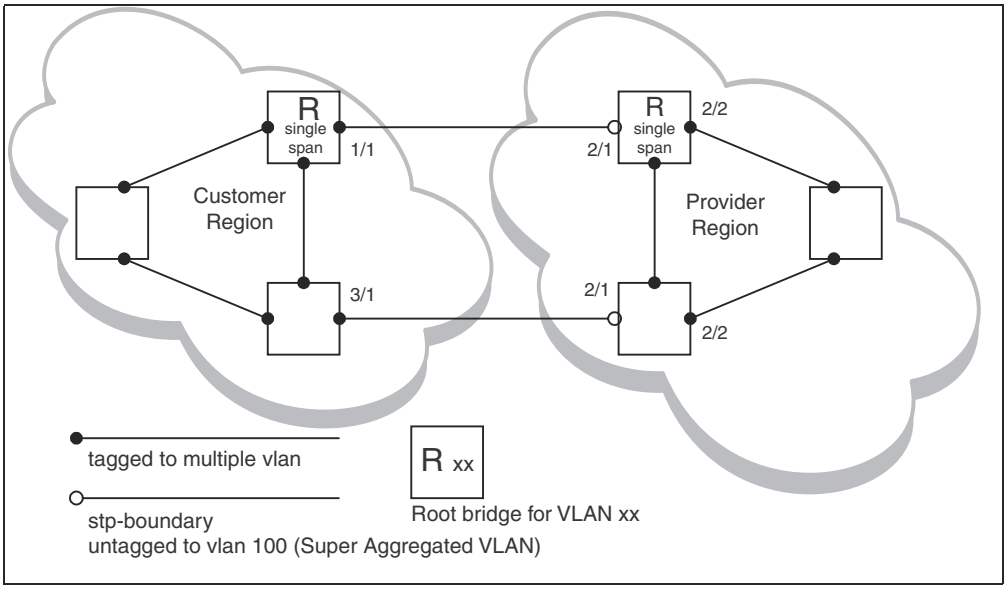


In this setup, the customer network is running a single spanning tree for VLANs 10 and 20. The traffic from VLAN 10 and 20 will be carried, or aggregated by VLAN 100 at the SP's network. The main difference between this scenario and the previous tow scenarios is that all traffic at the customer's network now follows the same path, having the same STP root bridge in all VLANs. Therefore, the customer network will not have the ability to maximize network utilization on all its links. On the other hand, loop-free, non-blocking topology is still separately maintained by the customer network's single spanning tree and the SP's per-VLAN spanning tree on VLAN 100.

Customer and SP use single STP

Figure 25 shows an example of SuperSpan where the customer network and SP both use Single STP.

FIGURE 25 Customer and SP using Single STP



In this setup, both the customer and SP networks are running a single spanning tree at Layer 2. The traffic from VLAN 10 and 20 will be carried, or aggregated by VLAN 100 at the SP network as in the previous scenario. Loop-free, non-blocking topology is still separately maintained by the customer's single spanning tree and the SP's single spanning tree.

Configuring SuperSpan

To configure the PowerConnect for SuperSpan:

- Configure each interface on the PowerConnect that is connected to customer equipment as a boundary interface. This step enables the interface to convert the destination MAC address in the customer's BPDUs.

  The software requires you to specify a SuperSpan customer ID when configuring the boundary interface. Use an ID from 1 – 65535. The customer ID uniquely identifies the customer. Use the same customer ID for each SP interface with the same customer. When tunneling BPDUs through the network, the PowerConnect devices use the customer ID to ensure that BPDUs are forwarded only to the customer's devices, and not to other customers' devices.

- Globally enable SuperSpan. This step enables the Preforwarding state.

Configuring a boundary interface

To configure the boundary interfaces on SP 1 in Figure 20 on page 388, enter the following commands.

```
NetIron(config)# interface 1/1
NetIron(config-if-e1000-1/1)# stp-boundary 1
NetIron(config)# interface 1/2
NetIron(config-if-e1000-1/2)# stp-boundary 2
```

These commands configure two interfaces on the PowerConnect as SuperSpan boundary interfaces. Interface
1/1 is a boundary interface with customer 1. Interface 1/2 is a boundary interface with customer 2. Each boundary interface is associated with a number, which is the SuperSpan ID. The SuperSpan ID identifies the instance of SuperSpan you are associating with the interface. Use the same SuperSpan ID for each boundary interface with the same customer. Use a different SuperSpan ID for each customer. For example, use SuperSpan ID 1 for all the boundary interfaces with customer 1 and use SuperSpan ID 2 for all boundary interfaces with customer 2.

Syntax: [no] stp-boundary <*num*>

The <*num*> parameter specifies the SuperSpan ID. Possible values: 1 – 65535.

To configure the boundary interfaces on SP 2 in Figure 20 on page 388, enter the following commands.

```
NetIron(config)# interface 2/1
NetIron(config-if-e1000-2/1)# stp-boundary 1
NetIron(config)# interface 2/2
NetIron(config-if-e1000-2/2)# stp-boundary 2
```

Enabling SuperSpan

After you configure the SuperSpan boundary interfaces, enable SuperSpan. You can enable SuperSpan globally or on an individual VLAN level. If you enable the feature globally, the feature is enabled on all VLANs.

NOTE
If you enable the feature globally, then create a new VLAN, the new VLAN inherits the global SuperSpan state. For example, if SuperSpan is globally enabled when you create a VLAN, SuperSpan also is enabled in the new VLAN.

You also can change the length of the preforwarding state.

To globally enable SuperSpan, enter the following command.

```
NetIron(config)# super-span
```

Syntax: [no] super-span [preforward-delay <*secs*>**]**

The <*secs*> parameter specifies the length of the preforwarding state. You can specify from 3 – 15 seconds. The default is 5 seconds.

SuperSpan is enabled in all VLANs on the PowerConnect. To disable SuperSpan in an individual VLAN, enter commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# no super-span
```

Syntax: [no] super-span

Displaying SuperSpan information

To display the boundary interface configuration and BPDU statistics, enter the following command.

```
NetIron(config)# show super-span
CID 1 Boundary Ports:
     Port  Customer    Tunnel
           BPDU Rx     BPDU Rx
     1/1   1            1
     1/2   0            0
     Total 1            1

CID 2 Boundary Ports:
     Port  Customer    Tunnel
           BPDU Rx     BPDU Rx
     2/1   0            3
     2/2   0            0
     Total 0            3
```

In this example, the PowerConnect has two SuperSpan customer IDs.

Syntax: show superspan [cid <*num*>**]**

The **cid** <*num*> parameter specifies a SuperSpan customer ID. If you do not specify a customer ID, information for all the customer IDs configured on the PowerConnect is shown.

This command shows the following information.

TABLE 85 CLI display of SuperSpan customer ID information

This field...	Displays...
CID	The SuperSpan customer ID number.
Port	The boundary port number.
Customer BPDU Rx	The number of BPDUs received from the client spanning tree.
Tunnel BPDU Rx	The number of BPDUs received from the SuperSpan tunnel.

To display general STP information, refer to "Displaying STP information" on page 380.

PVST or PVST+ compatibility

Dell's support for Cisco's Per VLAN Spanning Tree plus (PVST+) allows the PowerConnect to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices[1]. Ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected.

When it is configured for MSTP, the PowerConnect can interoperate with PVST.

1. Cisco user documentation for PVST or PVST+ refers to the IEEE 802.1Q spanning tree as the ***Common Spanning Tree (CST)***.

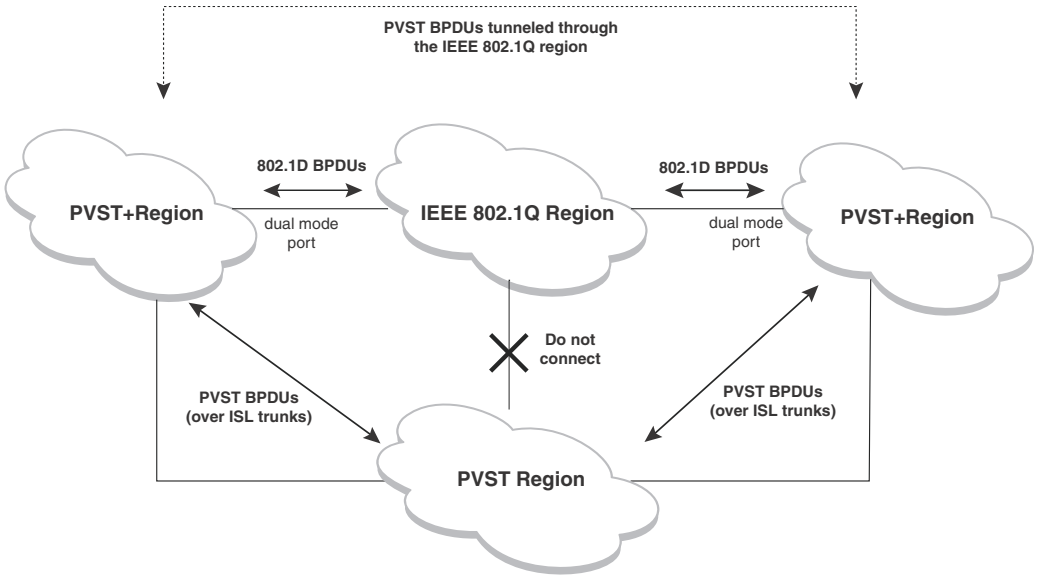
Overview of PVST and PVST+

Per VLAN Spanning Tree (PVST) is a Cisco proprietary protocol that allows a Cisco device to have multiple spanning trees. The Cisco device can interoperate with spanning trees on other PVST devices but cannot interoperate with IEEE 802.1Q devices. An IEEE 802.1Q device has all its ports running a single spanning tree. ***PVST+*** is an extension of PVST that allows a Cisco device to also interoperate with devices that are running a single spanning tree (IEEE 802.1Q).

The PVST+ support allows the PowerConnect to interoperate with PVST spanning trees and the IEEE 802.1Q spanning tree at the same time.

IEEE 802.1Q and PVST regions cannot interoperate directly but can interoperate indirectly through PVST+ regions. PVST BPDUs are tunneled through 802.1Q regions, while PVST BPDUs for VLAN 1 (the IEEE 802.1Q VLAN) are processed by PVST+ regions. Figure 26 shows the interaction of IEEE 802.1Q, PVST, and PVST+ regions.

FIGURE 26 Interaction of IEEE 802.1Q, PVST, and PVST+ regions



VLAN Tags and dual mode

The **dual-mode** feature enables the port to send and receive both tagged and untagged frames on a port. When the dual-mode feature is enabled, the port is an untagged member of one of its VLANs and is at the same time a tagged member of all its other VLANs. The untagged frames are supported on the port's **Port Native VLAN**.

To interoperate with other vendors, the dual-mode feature must be enabled on the port. Some vendors use VLAN 1 by default to support the IEEE 802.1Q based standard spanning tree protocols such as 802.1d and 802.1w for sending the untagged frames on VLAN 1. On Dell FastIron switches by default, the Port Native VLAN is the same as the device's **Default VLAN1**, which by default is VLAN 1. Thus, to support IEEE 802.1Q in a typical configuration, the port must be able to send and receive untagged frames for VLAN 1 and tagged frames for the other VLANs and interoperate with the vendors also using VLAN 1. If you want to use tagged frames on VLAN 1, you can change the

default VLAN ID to an ID other than 1. You also can specify the VLAN on which you want the port to send and receive untagged frames (the Port Native VLAN). The Port Native VLAN ID does not need to be the same as the Default VLAN. Make sure that untagged (Native) VLAN is also changed on the interoperating vendor side to match with that on the Dell side.

To support the IEEE 802.1Q with non-standard proprietary protocols such as PVST and PVST+, a port must always send and receive untagged frames on VLAN 1 on both sides. In that case, enable the dual-mode 1 feature to allow untagged BPDUs on VLAN 1and use Native VLAN 1 on the interoperating vendor side. You should not use VLAN 1 for tagged frames in this case.

NOTE
Support for the IEEE 802.1Q spanning tree always uses VLAN 1, regardless of whether the PowerConnect devices are configured to use tagged or untagged frames on the VLAN.

Enabling PVST+ support

PVST+ support is automatically enabled when the port receives a PVST BPDU. You can manually enable the support at any time or disable the support if desired.

If you want a tagged port to also support IEEE 802.1Q BPDUs, you need to enable the dual-mode feature on the port. The dual-mode feature is disabled by default and must be enabled manually.

A port that is in PVST+ compatibility mode due to auto-detection reverts to the default MSTP mode when one of the following events occurs:

- The link is disconnected or broken
- The link is administratively disabled
- The link is disabled by interaction with the link-keepalive protocol

This allows a port that was originally interoperating with PVST+ to revert to multiple spanning tree when connected to a PowerConnect.

Enabling PVST+ support manually

To immediately enable PVST+ support on a port, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# pvst-mode
```

Syntax: [no] pvst-mode

NOTE
If you disable PVST+ support, the software still automatically enables PVST+ support if the port receives a BPDU with PVST+ format.

Displaying PVST+ support information

To display PVST+ information for ports on a PowerConnect, enter the following command at any level of the CLI.

```
NetIron(config)# show span pvst-mode
PVST+ Enabled on:
Port      Method
1/1       Set by configuration
1/2       Set by configuration
2/10      Set by auto-detect
3/12      Set by configuration
4/24      Set by auto-detect
```

Syntax: show span pvst-mode

This command displays the following information.

TABLE 86 CLI display of PVST+ information

This field...	Displays...
Port	The port number. **NOTE:** The command lists information only for the ports on which PVST+ support is enabled.
Method	The method by which PVST+ support was enabled on the port. The method can be one of the following: • Set by configuration – You enabled the support. • Set by auto-detect – The support was enabled automatically when the port received a PVST+ BPDU.

Configuration examples

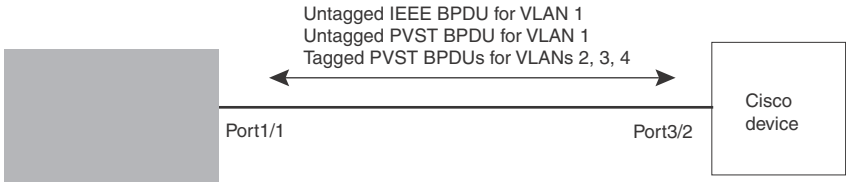
The examples use two common configurations:

- Untagged IEEE 802.1Q BPDUs on VLAN 1 and tagged PVST+ BPDUs on other VLANs
- Tagged IEEE 802.1Q BPDUs on VLAN 1 and untagged BPDUs on another VLAN

Tagged port using default VLAN 1 as its port native VLAN

In Figure 27, a PVST+ configuration uses VLAN 1 as the untagged default VLAN and VLANs 2, 3, and 4 as tagged VLANs.

FIGURE 27 Default VLAN 1 for untagged BPDUs



To implement this configuration, enter the following commands on the PowerConnect.

```
NetIron(config)# vlan-group 1 vlan 2 to 4
NetIron(config-vlan-group-1)# tagged ethernet 1/1
NetIron(config-vlan-group-1)# exit
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# pvst-mode
```

These commands configure a VLAN group containing VLANs 2, 3, and 4, add port 1/1 as a tagged port to the VLANs, and enable the dual-mode feature and PVST+ support on the port. The dual-mode feature allows the port to send and receive untagged frames for the default VLAN (VLAN 1 in this case) in addition to tagged frames for VLANs 2, 3, and 4. Enabling the PVST+ support ensures that the port is ready to send and receive PVST+ BPDUs. If you do not manually enable PVST+ support, the support is not enabled until the port receives a PVST+ BPDU.

The configuration leaves the default VLAN and the port's native VLAN unchanged. The default VLAN is 1 and the port's Port Native VLAN also is 1. The dual-mode feature supports untagged frames on the default VLAN only. Thus, port 1/1 can send and receive untagged BPDUs for VLAN 1 and can send and receive tagged BPDUs for the other VLANs.

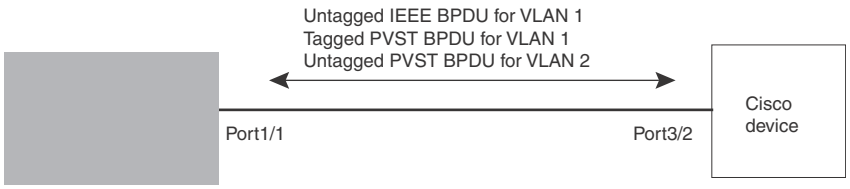
Port 1/1 will process BPDUs as follows:

- Process IEEE 802.1Q BPDUs for VLAN 1.
- Process tagged PVST BPDUs for VLANs 2, 3, and 4.
- Drop untagged PVST BPDUs for VLAN 1.

Untagged port using VLAN 2 as port native VLAN

In Figure 28, a port's Port Native VLAN is not VLAN 1. In this case, VLAN 1 uses tagged frames and VLAN 2 uses untagged frames.

FIGURE 28 Port Native VLAN 2 for untagged BPDUs



To implement this configuration, enter the following commands on the PowerConnect.

```
NetIron(config)# default-vlan-id 4000
NetIron(config)# vlan 1
NetIron(config-vlan-1)# tagged ethernet 1/1
NetIron(config-vlan-1)# exit
NetIron(config)# vlan 2
NetIron(config-vlan-2)# untagged ethernet 1/1
NetIron(config-vlan-2)# exit
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# pvst-mode
NetIron(config-if-e10000-1/1)# exit
```

These commands change the default VLAN ID, configure port 1/1 as a tagged member of VLANs 1 and 2, and enable PVST+ support on port 1/1. Since VLAN 1 is tagged in this configuration, the default VLAN ID must be changed from VLAN 1 to another VLAN ID. Changing the default VLAN ID from 1 allows the port to process tagged frames for VLAN 1. VLAN 2 is the port native VLAN. The port processes untagged frames and untagged PVST BPDUs on VLAN 2.

Port 1/1 will process BPDUs as follows:

- Process IEEE 802.1Q BPDUs for VLAN 1.
- Process untagged PVST BPDUs for VLAN 2.
- Drop tagged PVST BPDUs for VLAN 1.

Note that when VLAN 1 is not the default VLAN, the ports must have an untagged VLAN enabled in order to process IEEE 802.1Q BPDUs.

For example, the following configuration is incorrect.

```
NetIron(config)# default-vlan-id 1000
NetIron(config)# vlan 1
NetIron(config-vlan-1)# tagged ethernet 1/1 to 1/2
NetIron(config-vlan-1)# exit
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# pvst-mode
NetIron(config-if-e10000-1/1)# exit
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e10000-1/2)# pvst-mode
NetIron(config-if-e10000-1/2)# exit
```

In the configuration above, all PVST BPDUs associated with VLAN 1 would be discarded. Since IEEE BPDUs associated with VLAN 1 are untagged, they are discarded because the ports in VLAN 1 are tagged. Effectively, the BPDUs are never processed by the Spanning Tree Protocol. STP assumes that there is no better bridge on the network and sets the ports to FORWARDING. This could cause a Layer 2 loop.

The following configuration is correct.

```
NetIron(config)# default-vlan-id 1000
NetIron(config)# vlan 1
NetIron(config-vlan-1)# tagged ethernet 1/1 to 1/2
NetIron(config-vlan-1)# exit
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# pvst-mode
NetIron(config-if-e10000-1/1)# exit
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e10000-1/2)# pvst-mode
NetIron(config-if-e10000-1/2)# exit
```

Setting the ports as dual-mode ensures that the untagged IEEE 802.1Q BPDUs reach the VLAN 1 instance.

802.1s Multiple Spanning Tree Protocol

Multiple Spanning Tree Protocol (MSTP) as defined in IEEE 802.1s allows you to configure multiple STP instances. This will allow several VLANs to be mapped to a reduced number of spanning-tree instances. This ensures loop-free topology for 1 or more VLANs that have the same Layer 2 topology.

NOTE
In addition to the features described in this chapter, Root Guard and BPDU Guard are supported. Refer to "Root Guard" on page 375 and "BPDU Guard" on page 377 for details.

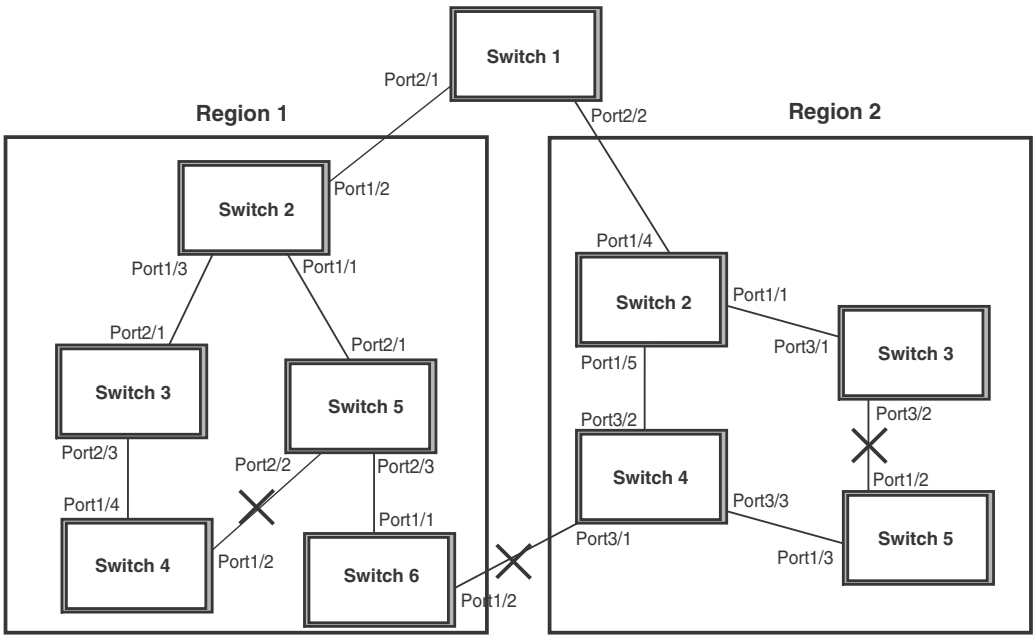
Multiple Spanning-Tree regions

Using MSTP, the entire network runs a common instance of RSTP. Within that common instance, one or more VLANs can be individually configured into distinct regions. The entire network runs the common spanning tree instance (CST) and the regions run a local instance. The local instance is known as Internal Spanning Tree (IST). The CST treats each instance of IST as a single bridge. Consequently, ports are blocked to prevent loops that might occur within an IST and also throughout the CST. In addition, MSTP can coexist with individual devices running STP or RSTP in the Common and Internal Spanning Trees instance (CIST). With the exception of the provisions for multiple instances, MSTP operates exactly like RSTP.

For example, in Figure 29 a network is configured with two regions: Region 1 and Region 2. The entire network is running an instance of CST. Each of the regions is running an instance of IST. In addition, this network contains Switch 1 running RSTP that is not configured in a region and consequently is running in the CIST instance. In this configuration, the regions are each regarded as a single bridge to the rest of the network, as is Switch 1. The CST prevents loops from occurring across the network. Consequently, a port is blocked at port 1/2 of switch 4.

Additionally, loops must be prevented in each of the IST instances. Within the IST Region 1, a port is blocked at port 1/2 of switch 4 to prevent a loop in that region. Within Region 2, a port is blocked at port 3/2 of switch 3 to prevent a loop in that region.

FIGURE 29 MSTP configured network



The following definitions describe the STP instances that define an MSTP configuration:

Common Spanning (CST) – MSTP runs a single instance of spanning tree, called the Common Spanning Tree (CST), across all the bridges in a network. This instance treats each region as a single bridge. It all other ways, it operates exactly like Rapid Spanning Tree (RSTP).

Internal Spanning Tree (IST) – Instances of spanning tree that operate within a defined region are called ISTs (Internal Spanning Tree).

Common and Internal Spanning Trees (CIST) – This is the default MSTP instance 0. It contains all of the ISTs and all bridges that are not formally configured into a region. This instance interoperates with bridges running legacy STP and RSTP implementations.

Multiple Spanning Tree Instance (MSTI) – The MSTI is identified by an MST identifier (MSTid) value between 1 and 4094. This defines an individual instance of an IST. One or more VLANs can be assigned to an MSTI. A VLAN cannot be assigned to multiple MSTIs.

MSTP Region – These are clusters of bridges that run multiple instances of the MSTP protocol. Multiple bridges detect that they are in the same region by exchanging their configuration (instance to VLAN mapping), name, and revision-level. Therefore, if you need to have two bridges in the same region, the two bridges must have identical configurations, names, and revision-levels.

Configuring MSTP

To configure a switch for MSTP, you could configure the name and the revision on each switch that is being configured for MSTP. This name is unique to each switch. You must then create an MSTP Instance and assign an ID. VLANs are then assigned to MSTP instances. These instances must be configured on all switches that interoperate with the same VLAN assignments. Port cost, priority and global parameters can then be configured for individual ports and instances. In addition, operational edge ports and point-to-point links can be created and MSTP can be disabled on individual ports.

MSTP can be configured on a router with MRP however, they are mutually exclusive on a specific VLAN. Also, MSTP can be configured on a port that is part of a LAG following the same rules as used for STP and RSTP.

Each of the commands used to configure and operate MSTP are described in the following:

- "Setting the MSTP name"
- "Setting the MSTP revision number"
- "Configuring an MSTP instance"
- "Configuring port priority and port path cost"
- "Configuring bridge priority for an MSTP instance"
- "Setting the MSTP global parameters"
- "Setting ports to be operational edge ports"
- "Setting point-to-point link"
- "Disabling MSTP on a port"
- "Forcing ports to transmit an MSTP BPDU"
- "Enabling MSTP on a switch"

Setting the MSTP name

Each switch that is running MSTP is configured with a name. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions. By default, the name is the MAC address of the device.

To configure an MSTP name, use a command such as the following at the Global Configuration level.

```
NetIron(config)# mstp name mstp1
```

Syntax: [no] mstp name <*name*>

The **name** parameter defines an ASCII name for the MSTP configuration. The default name is the MAC address of the switch expressed as a string.

Setting the MSTP revision number

Each switch that is running MSTP is configured with a revision number. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions.

To configure an MSTP revision number, use a command such as the following at the Global Configuration level.

```
NetIron(config)# mstp revision 4
```

Syntax: [no] mstp revision <*revision-number*>

The **revision** parameter specifies the revision level for MSTP that you are configuring on the switch. It can be a number from 0 and 65535.

Configuring an MSTP instance

An MSTP instance is configured with an MSTP ID for each region. Each region can contain one or more VLANs. To configure an MSTP instance and assign a range of VLANs, use a command such as the following at the Global Configuration level.

```
NetIron(config) # mstp instance 7 vlan 4 to 7
```

Syntax: [no] mstp instance <*instance-number*> [vlan <*vlan-id*> | vlan-group <*group-id*>]

The **instance** parameter defines the number for the instance of MSTP that you are configuring. The maximum number of instances that can be configured is 16.

The **vlan** parameter assigns one or more VLANs or a range of VLANs to the instance defined in this command.

The **vlan-group** parameter assigns one or more VLAN groups to the instance defined in this command.

Configuring port priority and port path cost

Priority and path cost can be configured for a specified instance. To configure an MSTP instance, use a command such as the following at the Global Configuration level.

```
NetIron(config)# mstp instance 7 ethernet 3/1 priority 32 path-cost 200
```

Syntax: [no] mstp instance <*instance-number*> ethernet <*slot/port*> priority <*port-priority*> path-cost <*cost*>

The <*instance-number*> variable is the number of the instance of MSTP that you are configuring priority and path cost for.

The **ethernet** <*slot/port*> parameter specifies a port within a VLAN. The priority and path cost configured with this command will apply to VLAN that the port is contained within.

You can set a **priority** to the port that gives it forwarding preference over lower priority instances within a VLAN or on the switch. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 240 in increments of 16. The default value is 128.

A **path-cost** can be assigned to a port to bias traffic towards or away from a path during periods of rerouting. Possible values are 1 - 200000000.

Configuring bridge priority for an MSTP instance

Priority can be configured for a specified instance. To configure priority for an MSTP instance, use a command such as the following at the Global Configuration level.

```
NetIron(config)# mstp instance 1 priority 8192
```

Syntax: [no] mstp instance <*instance-number*> **priority** <*priority-value*>

The <*instance-number*> variable is the number for the instance of MSTP that you are configuring.

You can set a **priority** to the instance that gives it forwarding preference over lower priority instances within a VLAN or on the switch. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 61440 in increments of 4096. The default value is 32768.

Setting the MSTP global parameters

MSTP has many of the options available in RSTP as well as some unique options. To configure MSTP Global parameters for all instances on a switch.

```
NetIron(config)# mstp force-version 0 forward-delay 10 hello-time 4 max-age 12
max-hops 9
```

Syntax: [no] mstp force-version <*mode-number*> **forward-delay** <*value*> **hello-time** <*value*> **max-age** <*value*> **max-hops** <*value*>

The **force-version** parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following <*mode-number*> values:

- 0 – The STP compatibility mode. Only STP BPDUs will be sent. This is equivalent to single STP.
- 2 – The RSTP compatibility mode. Only RSTP BPDUS will be sent. This is equivalent to single STP.
- 3 – MSTP mode. In this default mode, only MSTP BPDUS will be sent.

The **forward-delay** <*value*> specifies how long a port waits before it forwards an RST BPDU after a topology change. This can be a value from 4 – 30 seconds. The default is 15 seconds.

The **hello-time** <*value*> parameter specifies the interval between two hello packets. The parameter can have a value from 1 – 10 seconds. The default is 2 seconds.

The **max-age** <*value*> parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. You can specify a value from 6 – 40 seconds. The default value is 20 seconds.

The **max-hops** <*value*> parameter specifies the maximum hop count. You can specify a value from 1 – 40 hops. The default value is 20 hops.

Setting ports to be operational edge ports

You can define specific ports as edge ports for the region in which they are configured to connect to devices (such as a host) that are not running STP, RSTP, or MSTP. If a port is connected to an end device such as a PC, the port can be configured as an edge port. To configure ports as operational edge ports enter a command such as the following.

```
NetIron(config)# mstp admin-edge-port ethernet 3/1
```

Syntax: [no] mstp admin-edge-port ethernet <*slot/port*>

The <*slot/port*> parameter specifies a port or range of ports as edge ports in the instance they are configured in.

Setting point-to-point link

You can set a point-to-point link between ports to increase the speed of convergence. To create a point-to-point link between ports, use a command such as the following at the Global Configuration level.

```
NetIron(config)# mstp admin-pt2pt-mac ethernet 2/5 ethernet 4/5
```

Syntax: [no] mstp admin-pt2pt-mac ethernet <*slot/port*>

The <*slot/port*> parameter specifies a port or range of ports to be configured for point-to-point links to increase the speed of convergence.

Disabling MSTP on a port

To disable MSTP on a specific port, use a command such as the following at the Global Configuration level.

```
NetIron(config)# mstp disable 2/1
```

Syntax: [no] mstp disable <*slot/port*>

The <*slot/port*> variable specifies the location of the port that you want to disable MSTP for.

Forcing ports to transmit an MSTP BPDU

To force a port to transmit an MSTP BPDU, use a command such as the following at the Global Configuration level.

```
NetIron(config)# mstp force-migration-check ethernet 3/1
```

Syntax: [no] mstp force-migration-check ethernet <*slot/port*>

The <*slot/port*> variable specifies the port or ports that you want to transmit an MSTP BPDU from.

Enabling MSTP on a switch

To enable MSTP on your switch, use a command such as the following at the Global Configuration level.

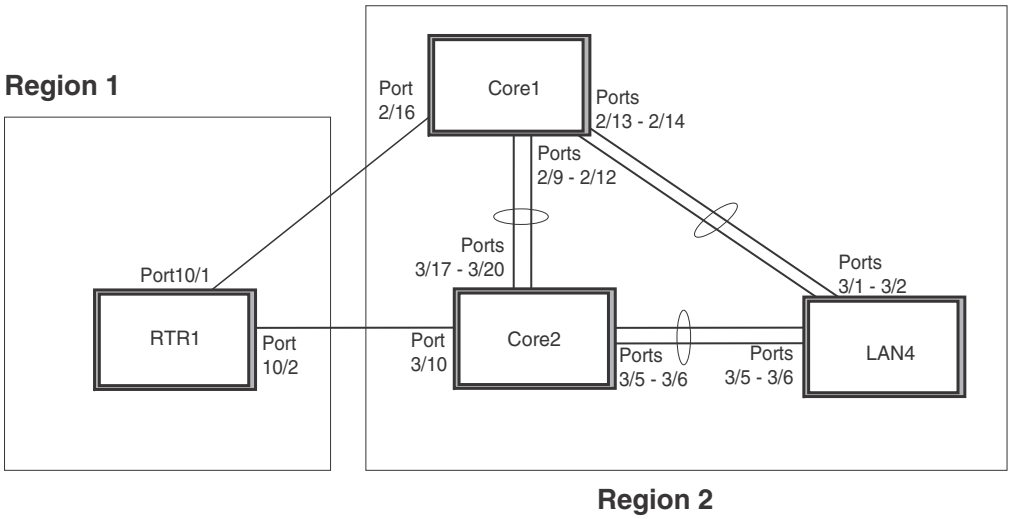
```
NetIron(config)# mstp start
```

Syntax: [no] start

Example

In Figure 30 four PowerConnect devices are configured in two regions. There are four VLANs in four instances in Region 2. Region 1 is in the CIST.

FIGURE 30 SAMPLE MSTP configuration



RTR1 configuration

```
NetIron(config-vlan-4093)tagged ethernet 10/1 to 10/2
NetIron(config-vlan-4093)exit
NetIron(config) mstp name Reg1
NetIron(config) mstp revision 1
NetIron(config) mstp instance 0 vlan 4093
NetIron(config) mstp admin-pt2pt-mac ethernet 10/1 to 10/2
NetIron(config) mstp start
NetIron(config) hostname RTR1
```

Core 1 configuration

```
NetIron(config) trunk switch ethernet 2/9 to 2/12 ethernet 2/13 to 2/14
NetIron(config-vlan-1) name DEFAULT-VLAN by port
NetIron(config-vlan-1)no spanning-tree
NetIron(config-vlan-1) exit
NetIron(config) vlan 20 by port
NetIron(config-vlan-20) tagged ethernet 2/9 to 2/14 ethernet 2/16
NetIron(config-vlan-20) no spanning-tree
NetIron(config-vlan-20) exit
NetIron(config) vlan 21 by port
NetIron(config-vlan-21) tagged ethernet 2/9 to 2/14 ethernet 2/16
NetIron(config-vlan-21) no spanning-tree
NetIron(config-vlan-21) exit
NetIron(config) vlan 22 by port
NetIron(config-vlan-22) tagged ethernet 2/9 to 2/14 ethernet 2/16
NetIron(config-vlan-22) no spanning-tree
NetIron(config-vlan-22) exit
NetIron(config) vlan 23 by port
NetIron(config) mstp name HR
NetIron(config) mstp revision 2
NetIron(config) mstp instance 20 vlan 20
NetIron(config) mstp instance 21 vlan 21
NetIron(config) mstp instance 22 vlan 22
```

```
NetIron(config) mstp instance 0 priority 8192
NetIron(config) mstp admin-pt2pt-mac ethernet 2/9 to 2/14
NetIron(config) mstp admin-pt2pt-mac ethernet 2/16
NetIron(config) mstp disable ethernet 2/240.

NetIron(config) mstp start
NetIron(config) hostname CORE1
```

Core2 configuration

```
NetIron(config) trunk switch ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
NetIron(config) vlan 1 name DEFAULT-VLAN by port
NetIron(config-vlan-1) no spanning-tree
NetIron(config-vlan-1) exit
NetIron(config) vlan 20 by port
NetIron(config-vlan-20) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
NetIron(config-vlan-20) no spanning-tree
NetIron(config-vlan-20) exit
NetIron(config) vlan 21 by port
NetIron(config-vlan-21) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
NetIron(config-vlan-21) no spanning-tree
NetIron(config-vlan-21) exit
NetIron(config) vlan 22 by port
NetIron(config-vlan-22) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
NetIron(config-vlan-22) no spanning-tree
NetIron(config-vlan-22) exit
NetIron(config) mstp name HR
NetIron(config) mstp revision 2
NetIron(config) mstp instance 20 vlan 20
NetIron(config) mstp instance 21 vlan 21
NetIron(config) mstp instance 22 vlan 22
NetIron(config) mstp admin-pt2pt-mac ethernet 3/17 to 3/20 ethernet 3/5 to 3/6
NetIron(config) mstp admin-pt2pt-mac ethernet 3/10
NetIron(config) mstp disable ethernet 3/7 ethernet 3/24
NetIron(config) mstp start
NetIron(config) hostname CORE2
```

LAN 4 configuration

```
NetIron(config) trunk switch ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
NetIron(config) vlan 1 name DEFAULT-VLAN by port
NetIron(config-vlan-1) no spanning-tree
NetIron(config-vlan-1) exit
NetIron(config) vlan 20 by port
NetIron(config-vlan-20) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
NetIron(config-vlan-20) no spanning-tree
NetIron(config) exit
NetIron(config) vlan 21 by port
NetIron(config-vlan-21) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
NetIron(config-vlan-21) no spanning-tree
NetIron(config-vlan-21) exit
NetIron(config) vlan 22 by port
NetIron(config-vlan-22) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
NetIron(config-vlan-22) no spanning-tree
NetIron(config) mstp config name HR
NetIron(config) mstp revision 2
NetIron(config) mstp instance 20 vlan 20
NetIron(config) mstp instance 21 vlan 21
```

```
NetIron(config) mstp instance 22 vlan 22
NetIron(config) mstp admin-pt2pt-mac ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
NetIron(config) mstp start
NetIron(config) hostname LAN4
```

Displaying MSTP statistics

MSTP statistics can be displayed using the commands shown below.

To display all general MSTP information, enter the following command.

```
NetIron(config)#show mstp
MSTP Instance 0 (CIST) - VLANs: 1
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19

Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876

MSTP Instance 1 - VLANs: 2
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20

Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                         ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

Syntax: show mstp <*instance-number*>

The <*instance-number*> variable specifies the MSTP instance that you want to display information for.

TABLE 87 Output from show MSTP

This field...	Displays...
MSTP Instance	The ID of the MSTP instance whose statistics are being displayed. For the CIST, this number is 0.
VLANs:	The number of VLANs that are included in this instance of MSTP. For the CIST this number will always be 1.
Bridge Identifier	The MAC address of the bridge.
Bridge MaxAge sec	Displays configured Max Age.
Bridge Hello sec	Displays configured Hello variable.
Bridge FwdDly sec	Displays configured FwdDly variable.
Bridge Hop cnt	Displays configured Max Hop count variable.

TABLE 87 Output from show MSTP (Continued)

This field...	Displays...
Root MaxAge sec	Max Age configured on the root bridge.
Root Hello sec	Hello interval configured on the root bridge.
Root FwdDly sec	FwdDly interval configured on the root bridge.
Root Hop Cnt	Current hop count from the root bridge.
Root Bridge	Bridge identifier of the root bridge.
ExtPath Cost	The configured path cost on a link connected to this port to an external MSTP region.
Regional Root Bridge	The Regional Root Bridge is the MAC address of the Root Bridge for the local region.
IntPath Cost	The configured path cost on a link connected to this port within the internal MSTP region.
Designated Bridge	The MAC address of the bridge that sent the best BPDU that was received on this port.
Root Port	Port indicating shortest path to root. Set to "Root" if this bridge is the root bridge.
Port Num	The port number of the interface.
Pri	The configured priority of the port. The default is 128.
PortPath Cost	Configured or auto detected path cost for port.
P2P Mac	Indicates if the port is configured with a point-to-point link: • **T** – The port is configured in a point-to-point link • **F** – The port is not configured in a point-to-point link
Edge	Indicates if the port is configured as an operational edge port: • **T** – indicates that the port is defined as an edge port. • **F** – indicates that the port is not defined as an edge port
Role	The current role of the port: • Master • Root • Designated • Alternate • Backup • Disabled
State	The port's current 802.1w state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled
Designated Cost	Port path cost to the root bridge.
Max Hop cnt	The maximum hop count configured for this instance.
Root Hop cnt	Hop count from the root bridge.

Displaying MSTP information for a specified instance

The following example displays MSTP information specified for an MSTP instance.

```
NetIron(config)#show mstp 1
MSTP Instance 1 - VLANs: 2
------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20

Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                          ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

Refer to Table 87 for details about the display parameters.

Displaying MSTP information for CIST instance 0

Instance 0 is the Common and Internal Spanning Tree Instance (CIST). When you display information for this instance there are some differences with displaying other instances. The following example displays MSTP information for CIST Instance 0.

```
NetIron(config)#show mstp 0
MSTP Instance 0 (CIST) - VLANs: 1
------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876
```

To display details about the MSTP configuration, enter the following command.

```
NetIron(config)#show mstp conf
MSTP CONFIGURATION
------------------
Name     : Reg1
Revision : 1
Version  : 3 (MSTP mode)
Status   : Started

Instance VLANs
-------- ----------------------------------------------------
0        4093
```

To display details about the MSTP that is configured on the device, enter the following command.

```
NetIron(config)#show mstp detail
MSTP Instance 0 (CIST) - VLANs: 4093
--------------------------------------------------------------------------------
Bridge: 800000b000c00000 [Priority 32768, SysId 0, Mac 00b000c00000]
FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 6/54 - Role: DESIGNATED - State: FORWARDING
PathCost 20000, Priority 128, OperEdge T, OperPt2PtMac F, Boundary T
Designated - Root 800000b000c00000, RegionalRoot 800000b000c00000,
Bridge 800000b000c00000, ExtCost 0, IntCost 0
ActiveTimers - helloWhen 1
MachineState - PRX-DISCARD, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
PRT-ACTIVE_PORT, PST-FORWARDING, TCM-INACTIVE
BPDUs - Rcvd MST 0, RST 0, Config 0, TCN 0
Sent MST 6, RST 0, Config 0, TCN 0
```

Refer to Table 87 for explanation about the parameters in the output.

Syntax: show mstp [<*mstp-id*> | **configuration** | **detail**] [| **begin** <*string*> | **exclude** <*string*> | **include** <*string*>]

Enter an MSTP ID for <*mstp-id*>.

Configuring STP under an ESI VLAN

STP can also be configured under a VLAN that is part of a user-configured ESI. For example, to enable spanning tree on a VLAN that is part of an ESI, configure the following commands.

```
NetIron(config)# esi customer1 encapsulation cvlan
NetIron(config-esi-customer1)# vlan 100
NetIron(config-esi-customer1-vlan-100)# spanning-tree
```

Configuration considerations:

The configuration considerations are as follows:

- MSTP can only be configured under the default ESI. MSTP cannot be configured for VLANs that are configured under a user-defined ESI.
- STP can be configured for VLANs with encapsulation type B-VLAN, S-VLAN or C-VLAN
- When STP or RSTP is configured for VLANs under an ESI, the MRP members must be part of the same ESI

Configuring Rapid Spanning Tree Protocol 12

The following Rapid Spanning Tree Protocol features are supported by NetIron MLX Series devices.

- Rapid Spanning Tree Protocol
- Edge Ports
- Point-to-Point Ports
- Convergence in a Simple Topology
- Convergence in a Complex RSTP Topology
- Compatibility of RSTP with 802.1D
- RSTP support under an ESI with support for B-VLANs, S-VLANs and C-VLANs

This chapter explains the IEEE 802.1W-2001Rapid Spanning Tree Protocols (RSTP) support on the PowerConnect.

NOTE
In addition to the features described in this chapter, Root Guard and BPDU Guard. Refer to "Root Guard" on page 375 and "BPDU Guard" on page 377 for details.

IEEE 802.1W-2001 RSTP provides rapid traffic reconvergence for point-to-point links within a few milliseconds (< 500 milliseconds), following the failure of a bridge or bridge port.

This reconvergence occurs more rapidly than the reconvergence provided by the IEEE 802.1D Spanning Tree Protocol or by RSTP Draft 3 because:

- STP requires a newly selected Root port to go through listening and learning stages before traffic convergence can be achieved. The STP traffic convergence time is calculated using the following formula:

  2 x FORWARD_DELAY + BRIDGE_MAX_AGE.

- Convergence in RSTP bridges is not based on any timer values. Rather, it is based on the explicit handshakes between Designated ports and their connected Root ports to achieve convergence in less than 500 milliseconds.

NOTE
The rapid convergence will not occur on ports connected to shared media devices, such as hubs. To take advantage of the rapid convergence provided by RSTP, make sure to explicitly configure all point-to-point links in a topology.

Bridges and bridge port roles

A bridge in an RSTP rapid spanning tree topology is assigned as the root bridge if it has the highest priority (lowest bridge identifier) in the topology. Other bridges are referred to as non-root bridges.

Unique roles are assigned to ports on the root and non-root bridges. Role assignments are based on the following information contained in the BPDU (RSTp packet):

- Root bridge ID
- Path cost value
- Transmitting bridge ID
- Designated port ID

RSTP algorithm uses this information to determine if the RST BPDU received by a port is superior to the RST BPDU that the port transmits. The two values are compared in the order as given above, starting with the Root bridge ID. The RST BPDU with a lower value is considered superior. The superiority and inferiority of the RST BPDU is used to assign a role to a port.

If the value of the received RST BPDU is the same as that of the transmitted RST BPDU, then the port ID in the RST BPDUs are compared. The RST BPDU with the lower port ID is superior. Port roles are then calculated appropriately.

The port's role is included in the BPDU that it transmits. The BPDU transmitted by an RSTP port is referred to as an RST BPDU, while it is operating in RSTP mode.

Ports can have one of the following roles:

- **Root** – Provides the lowest cost path to the root bridge from a specific bridge
- **Designated** – Provides the lowest cost path to the root bridge from a LAN to which it is connected
- **Alternate** – Provides an alternate path to the root bridge when the root port goes down
- **Backup** – Provides a backup to the LAN when the Designated port goes down
- **Disabled** – Has no role in the topology

Assignment of port roles

At system start-up, all RSTP-enabled bridge ports assume a Designated role. Once start-up is complete, RSTP algorithm calculates the superiority or inferiority of the RST BPDU that is received and transmitted on a port.

On a root bridge, each port is assigned a ***Designated port*** role, except for ports on the same bridge that are physically connected together. In these type of ports, the port that receives the superior RST BPDU becomes the ***Backup port***, while the other port becomes the ***Designated port***.

On non-root bridges, ports are assigned as follows:

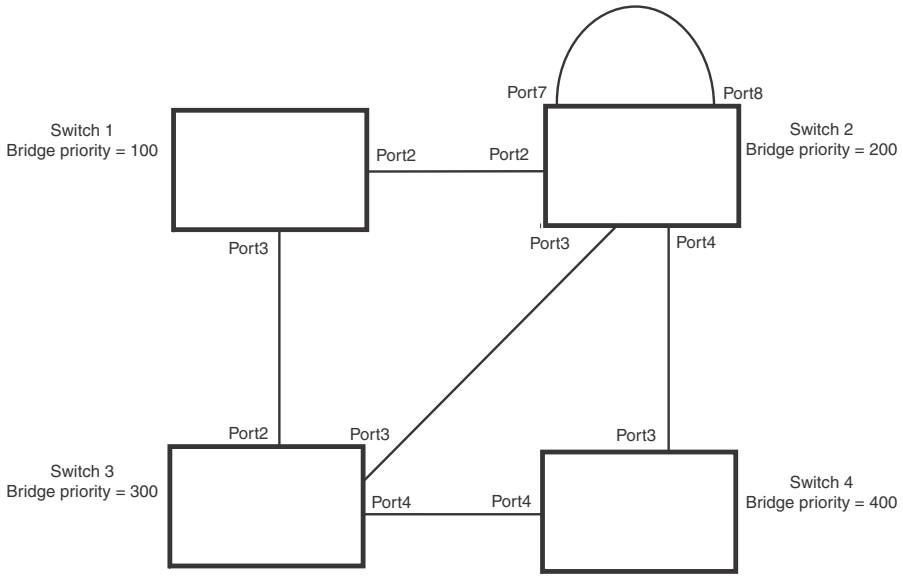
- The port that receives the RST BPDU with the lowest path cost from the root bridge becomes the ***Root port***.
- If two ports on the same bridge are physically connected, the port that receives the superior RST BPDU becomes the ***Backup port***, while the other port becomes the ***Designated port***.
- If a non-root bridge already has a Root port, then the port that receives an RST BPDU that is superior to those it can transmit becomes the ***Alternate port***.
- If the RST BPDU that a port receives is inferior to the RST BPDUs it transmits, then the port becomes a ***Designated port***.
- If the port is down or if RSTP is disabled on the port, that port is given the role of ***Disabled port***. Disabled ports have no role in the topology. However, if RSTP is enabled on a port with a link down and the link of that port comes up, then that port assumes one of the following port roles: Root, Designated, Alternate, or Backup.

The following example (Figure 31) explains role assignments in a simple RSTP topology.

NOTE
All examples in this document assume that all ports in the illustrated topologies are point-to-point links and are homogeneous (they have the same path cost value) unless otherwise specified.

The topology in Figure 31 contains four bridges. Switch 1 is the root bridge since it has the lowest bridge priority. Switch 2 through Switch 4 are non-root bridges.

FIGURE 31 Simple RSTP topology



Ports on Switch 1

All ports on Switch 1, the root bridge, are assigned Designated port roles.

Ports on Switch 2

Port2 on Switch 2 directly connects to the root bridge; therefore, Port2 is the Root port.

Switch 2's bridge priority value is superior to that of Switch 3 and Switch 4; therefore, the ports on Switch 2 that connect to Switch 3 and Switch 4 are given the Designated port role.

Furthermore, Port7 and Port8 on Switch 2 are physically connected. The RST BPDUs transmitted by Port7 are superior to those Port8 transmits. Therefore, Switch 2 is the Backup port and Port7 is the Designated port.

Ports on Switch 3

Port2 on Switch 3 directly connects to the Designated port on the root bridge; therefore, it assumes the Root port role.

The root path cost of the RST BPDUs received on Port4/Switch 3 is inferior to the RST BPDUs transmitted by the port; therefore, Port4/Switch 3 becomes the Designated port.

Similarly, Switch 3 has a bridge priority value inferior to Switch 2. Port3 on Switch 3 connects to Port 3 on Switch 2. This port will be given the Alternate port role, since a Root port is already established on this bridge.

Ports Switch 4

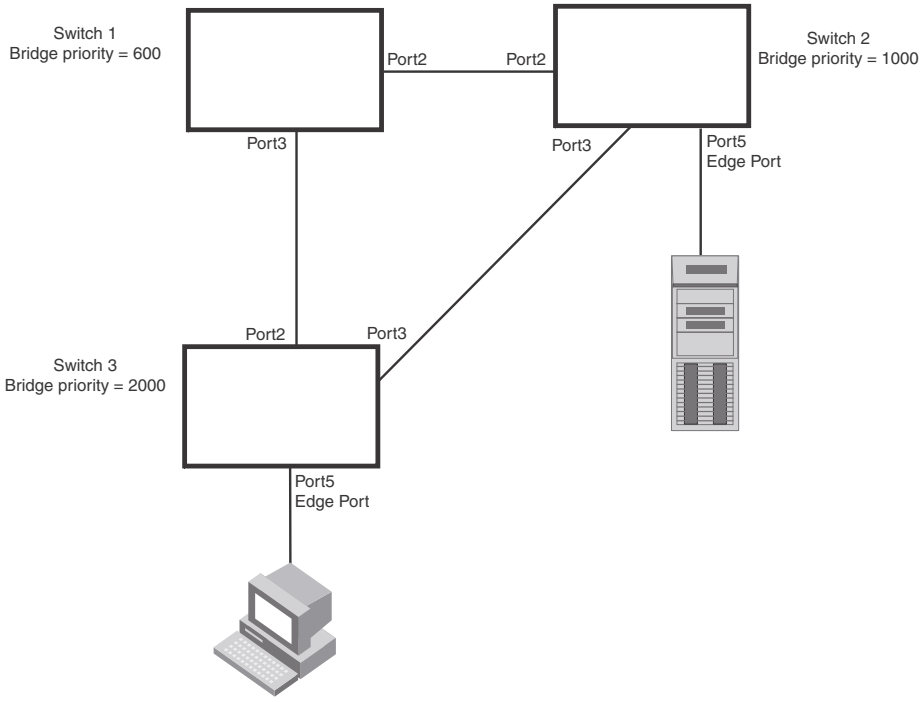
Switch 4 is not directly connected to the root bridge. It has two ports with superior incoming RST BPDUs from two separate LANs: Port3 and Port4. The RST BPDUs received on Port3 are superior to the RST BPDUs received on port 4; therefore, Port3 becomes the Root port and Port4 becomes the Alternate port.

Edge ports and Edge port roles

Dell's implementation of RSTP allows ports that are configured as Edge ports to be present in an RSTP topology. (Figure 32). Edge ports are ports of a bridge that connect to workstations or computers. Edge ports do not register any incoming BPDU activities.

Edge ports assume Designated port roles. Port flapping does not cause any topology change events on Edge ports since RSTP does not consider Edge ports in the spanning tree calculations.

FIGURE 32 Topology with edge ports



However, if any incoming RST BPDU is received from a previously configured Edge port, RSTP automatically makes the port as a non-edge port. This is extremely important to ensure a loop free Layer 2 operation since a non-edge port is part of the active RSTP topology.

The bridge detection state module can auto-detect an Edge port and a non-edge port. An administrator can also configure a port to be an Edge port. It is recommended that Edge ports are configured explicitly to take advantage of the Edge port feature, instead of allowing the protocol to auto-detect them.

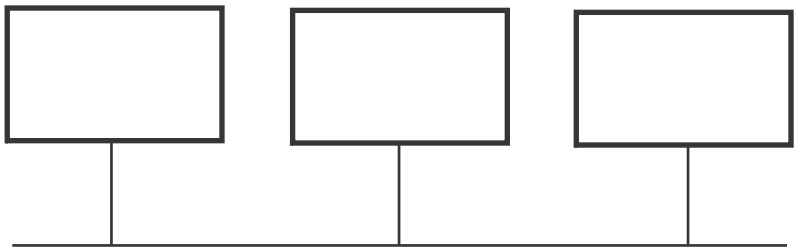
Point-to-point ports

To take advantage of the RSTP features, ports on an RSTP topology should be explicitly configured as point-to-point links. Shared media should not be configured as point-to-point links.

NOTE
Configuring shared media or non-point-to-point links as point-to-point links could lead to Layer 2 loops.

The topology in Figure 33 is an example of shared media that should not be configured as point-to-point links. In Figure 33, a port on a bridge communicates or is connected to at least two ports.

FIGURE 33 Example of shared media



Bridge port states

Ports roles can have one of the following states:

- **Forwarding** – RSTP is allowing the port to send and receive all packets.
- **Discarding** – RSTP has blocked data traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is forwarding. When a port is in this state, the port does not transmit or receive data frames, but the port does continue to receive RST BPDUs. This state corresponds to the listening and blocking states of 802.1D.
- **Learning** – RSTP is allowing MAC address entries to be added to the filtering database but does not permit forwarding of data frames. The device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
- **Disabled** – The port is not participating in RSTP. This can occur when the port is disconnected or RSTP is administratively disabled on the port.

A port on a non-root bridge with the role of Root port is always in a forwarding state. If another port on that bridge assumes the Root port role, then the old Root port moves into a discarding state as it assumes another port role.

A port on a non-root bridge with a Designated role starts in the discarding state. When that port becomes elected to the Root port role, RSTP quickly places it into a forwarding state. However, if the Designated port is an Edge port, then the port starts and stays in a forwarding state and it cannot be elected as a Root port.

A port with an Alternate or Backup role is always in a discarding state. If the port's role changes to Designated, then the port changes into a forwarding state.

If a port on one bridge has a Designated role and that port is connected to a port on another bridge that has an Alternate or Backup role, the port with a Designated role cannot be given a Root port role until two instances of the forward delay timer expires on that port.

Edge port and non-Edge port states

As soon as a port is configured as an Edge port, it goes into a forwarding state instantly (in less than 100 msec).

When the link to a port comes up and RSTP detects that the port is an Edge port, that port instantly goes into a forwarding state.

If RSTP detects that port as a non-edge port, the port goes into a forwarding state within four seconds of link up or after two hello timer expires on the port.

Changes to port roles and states

To achieve convergence in a topology, a port's role and state changes as it receives and transmits new RST BPDUs. Changes in a port's role and state constitute a topology change. Besides the superiority and inferiority of the RST BPDU, bridge-wide and per-port state machines are used to determine a port's role as well as a port's state. Port state machines also determine when port role and state changes occur.

State machines

The bridge uses the Port Role Selection state machine to determine if port role changes are required on the bridge. This state machine performs a computation when one of the following events occur:

- New information is received on any port on the bridge
- The timer expires for the current information on a port on the bridge

Each port uses the following state machines:

- **Port Information** – This state machine keeps track of spanning-tree information currently used by the port. It records the origin of the information and ages out any information that was derived from an incoming BPDU.
- **Port Role Transition** – This state machine keeps track of the current port role and transitions the port to the appropriate role when required. It moves the Root port and the Designated port into forwarding states and moves the Alternate and Backup ports into discarding states.

- **Port Transmit** – This state machine is responsible for BPDU transmission. It checks to ensure only the maximum number of BPDUs per hello interval are sent every second. Based on what mode it is operating in, it sends out either legacy BPDUs or RST BPDUs. In this document legacy BPDUs are also referred to as STP BPDUs.
- **Port Protocol Migration** – This state machine deals with compatibility with 802.1D bridges. When a legacy BPDU is detected on a port, this state machine configures the port to transmit and receive legacy BPDUs and operate in the legacy mode.
- **Topology Change** – This state machine detects, generates, and propagates topology change notifications. It acknowledges Topology Change Notice (TCN) messages when operating in 802.1D mode. It also flushes the MAC table when a topology change event takes place.
- **Port State Transition** – This state machine transitions the port to a discarding, learning, or forwarding state and performs any necessary processing associated with the state changes.
- **Port Timers** – This state machine is responsible for triggering any of the state machines described above, based on expiration of specific port timers.

In contrast to the 802.1D standard, the RSTP standard does not have any bridge specific timers. All timers in the CLI are applied on a per-port basis, even though they are configured under bridge parameters.

RSTP state machines attempt to quickly place the ports into either a forwarding or discarding state. Root ports are quickly placed in forwarding state when both of the following events occur:

- It is assigned to be the Root port.
- It receives an RST BPDU with a proposal flag from a Designated port. The proposal flag is sent by ports with a Designated role when they are ready to move into a forwarding state.

When a the role of Root port is given to another port, the old Root port is instructed to reroot. The old Root port goes into a discarding state and negotiates with its peer port for a new role and a new state. A peer port is the port on the other bridge to which the port is connected. For example, in Figure 34, Port1 of Switch 200 is the peer port of Port2 of Switch 100.

A port with a Designated role is quickly placed into a forwarding state if one of the following occurs:

- The Designated port receives an RST BPDU that contains an agreement flag from a Root port
- The Designated port is an Edge port

However, a Designated port that is attached to an Alternate port or a Backup port must wait until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state.

Backup ports are quickly placed into discarding states.

Alternate ports are quickly placed into discarding states.

A port operating in RSTP mode may enter a learning state to allow MAC address entries to be added to the filtering database; however, this state is transient and lasts only a few milliseconds, if the port is operating in RSTP mode and if the port meets the conditions for rapid transition.

Handshake mechanisms

To rapidly transition a Designated or Root port into a forwarding state, the Port Role Transition state machine uses handshake mechanisms to ensure loop free operations. It uses one type of handshake if no Root port has been assigned on a bridge, and another type if a Root port has already been assigned.

Handshake when no root port is elected

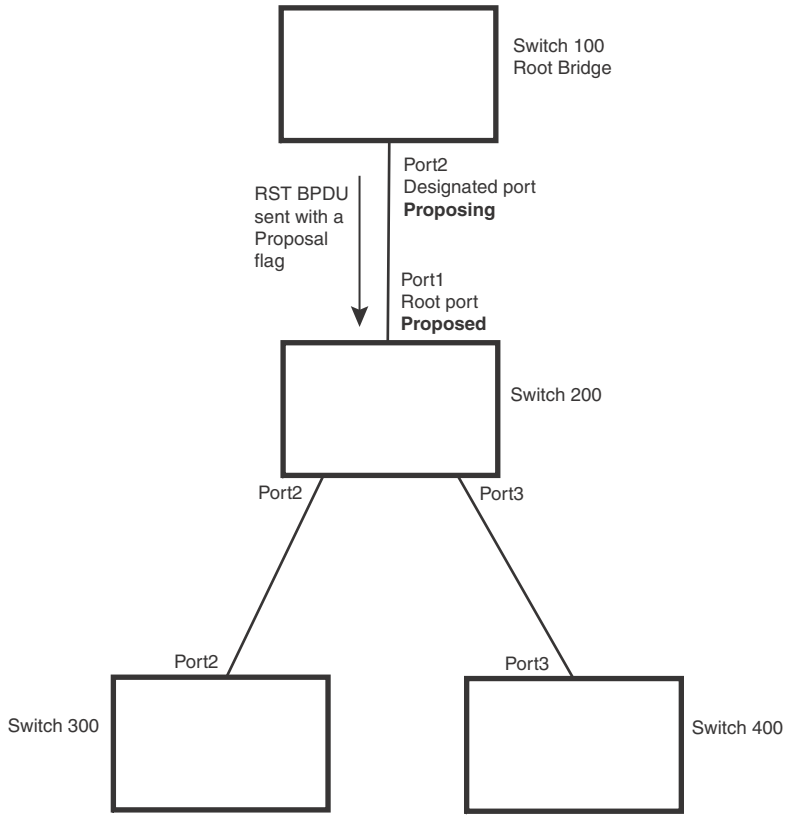
If a Root port has not been assigned on a bridge, RSTP uses the Proposing -> Proposed -> Sync -> Synced -> Agreed handshake:

- **Proposing** – The Designated port on the root bridge sends an RST BPDU packet to its peer port that contains a proposal flag. The proposal flag is a signal that indicates that the Designated port is ready to put itself in a forwarding state (Figure 34). The Designated port continues to send this flag in its RST BPDU until it is placed in a forwarding state (Figure 37) or is forced to operate in 802.1D mode. (Refer to "Compatibility of RSTP with 802.1D" on page 441)
- **Proposed** – When a port receives an RST BPDU with a proposal flag from the Designated port on its point-to-point link, it asserts the Proposed signal and one of the following occurs (Figure 34):
  - If the RST BPDU that the port receives is superior to what it can transmit, the port assumes the role of a Root port. (Refer to the section on "Bridges and bridge port roles" on page 413.)
  - If the RST BPDU that the port receives is inferior to what it can transmit, then the port is given the role of Designated port.

NOTE
Proposed will never be asserted if the port is connected on a shared media link.

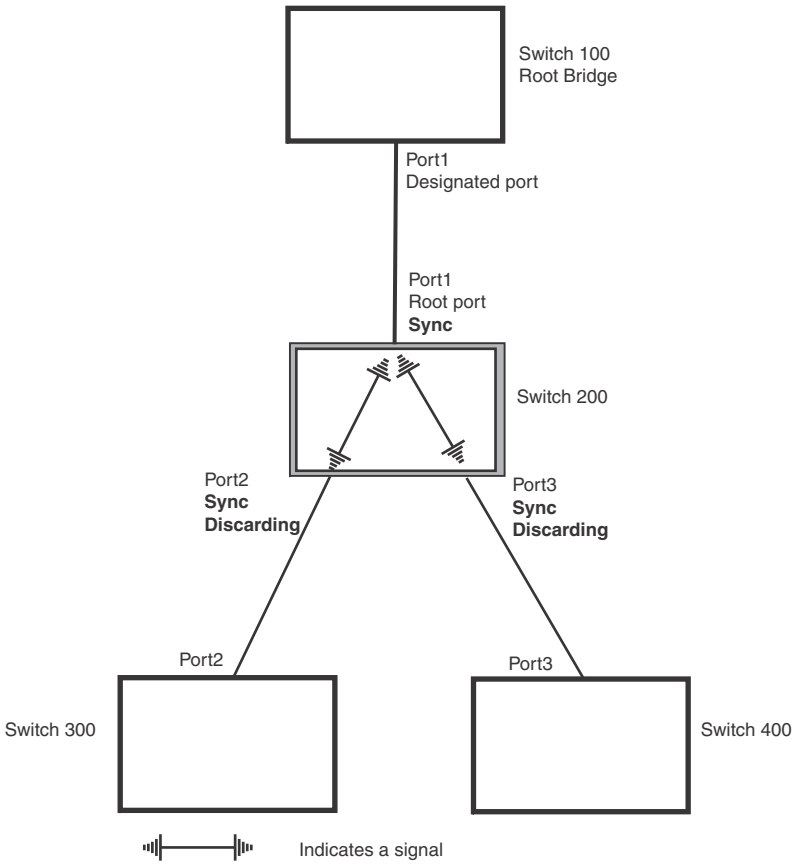
In Figure 34, Port3/Switch 200 is elected as the Root port

FIGURE 34 Proposing and proposed stage



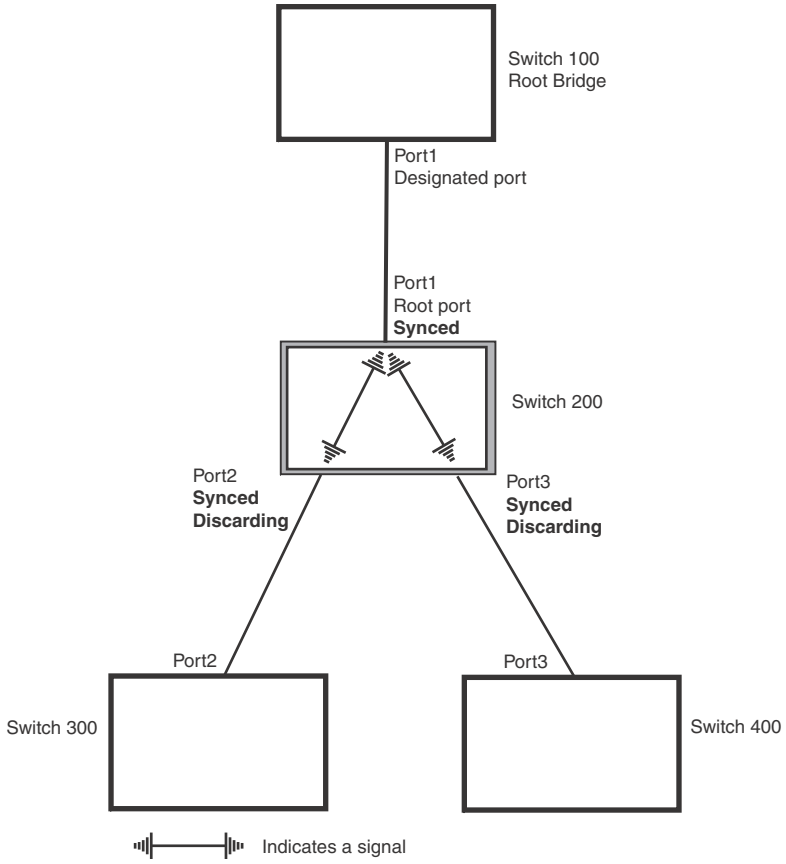
- **Sync** – Once the Root port is elected, it sets a sync signal on all the ports on the bridge. The signal tells the ports to synchronize their roles and states (Figure 35). Ports that are non-edge ports with a role of Designated port change into a discarding state. These ports have to negotiate with their peer ports to establish their new roles and states.

FIGURE 35 Sync stage



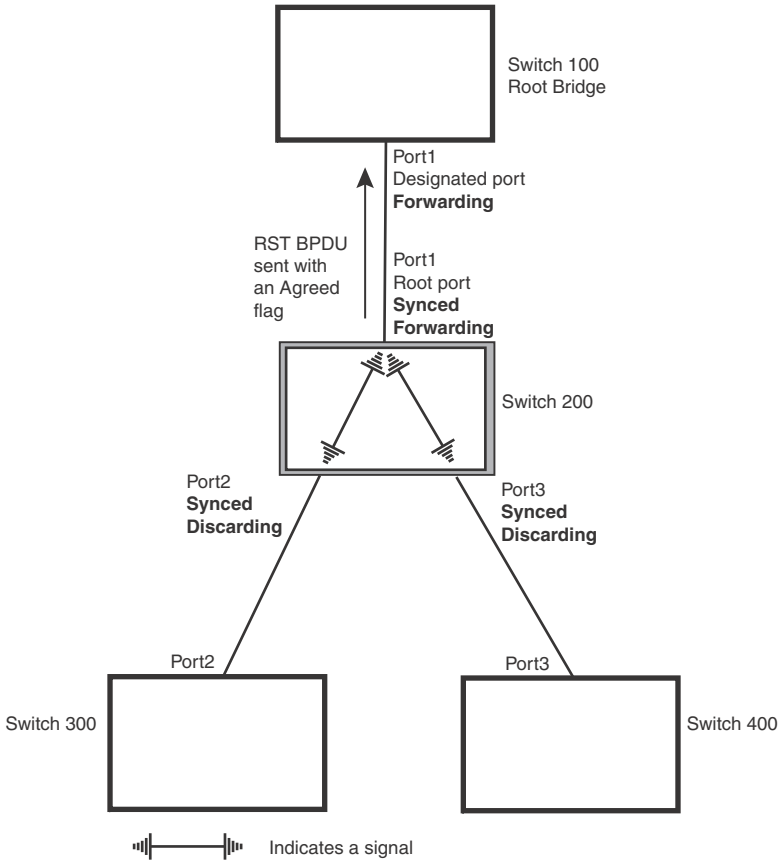
- **Synced** – Once the Designated port changes into a discarding state, it asserts a synced signal. Immediately, Alternate ports and Backup ports are synced. The Root port monitors the synced signals from all the bridge ports. Once all bridge ports asserts a synced signal, the Root port asserts its own synced signal (Figure 36).

FIGURE 36 Synced stage



- **Agreed** – The Root port sends back an RST BPDU containing an agreed flag to its peer Designated port and moves into the forwarding state. When the peer Designated port receives the RST BPDU, it rapidly transitions into a forwarding state.

FIGURE 37 Agree stage



At this point, the handshake mechanism is complete between Switch 100, the root bridge, and Switch 200.

Switch 200 updates the information on the Switch 200's Designated ports (Port2 and Port3) and identifies the new root bridge. The Designated ports send RST BPDUs, containing proposal flags, to their downstream bridges, without waiting for the hello timers to expire on them. This process starts the handshake with the downstream bridges.

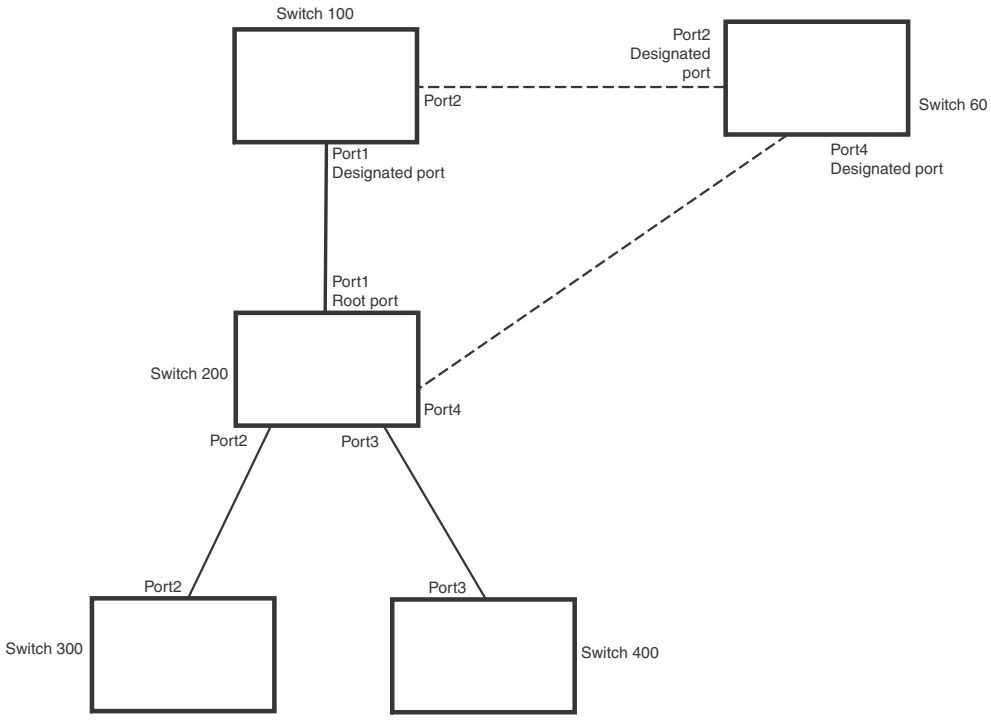
For example, Port2/Switch 200 sends an RST BPDU to Port2/Switch 300 that contains a proposal flag. Port2/Switch 300 asserts a proposed signal. Ports in Switch 300 then set sync signals on the ports to synchronize and negotiate their roles and states. Then the ports assert a synced signal and when the Root port in Switch 300 asserts it's synced signal, it sends an RST BPDU to Switch 200 with an agreed flag.

This handshake is repeated between Switch 200 and Switch 400 until all Designated and Root ports are in forwarding states.

Handshake when a root port has been elected

If a non-root bridge already has a Root port, RSTP uses a different type of handshake. For example, in Figure 38, a new root bridge is added to the topology.

FIGURE 38 Addition of a new root bridge

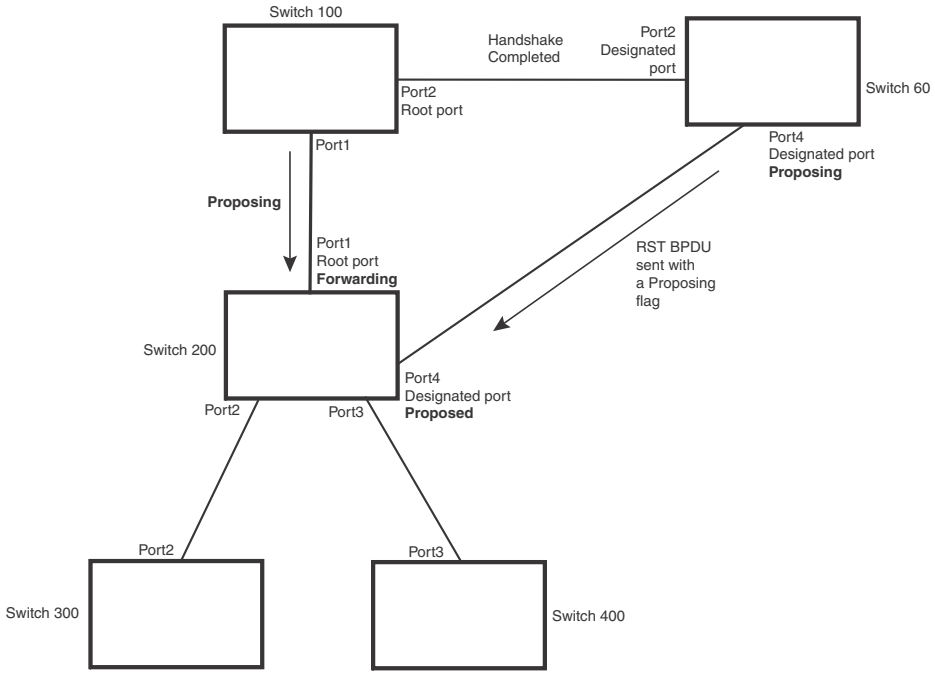


The handshake that occurs between Switch 60 and Switch 100 follows the one described in the previous section ("Handshake when no root port is elected" on page 420). The former root bridge becomes a non-root bridge and establishes a Root port (Figure 39).

However, since Switch 200 already had a Root port in a forwarding state, RSTP uses the Proposing -> Proposed -> Sync and Reroot -> Sync and Rerooted -> Rerooted and Synced -> Agreed handshake:

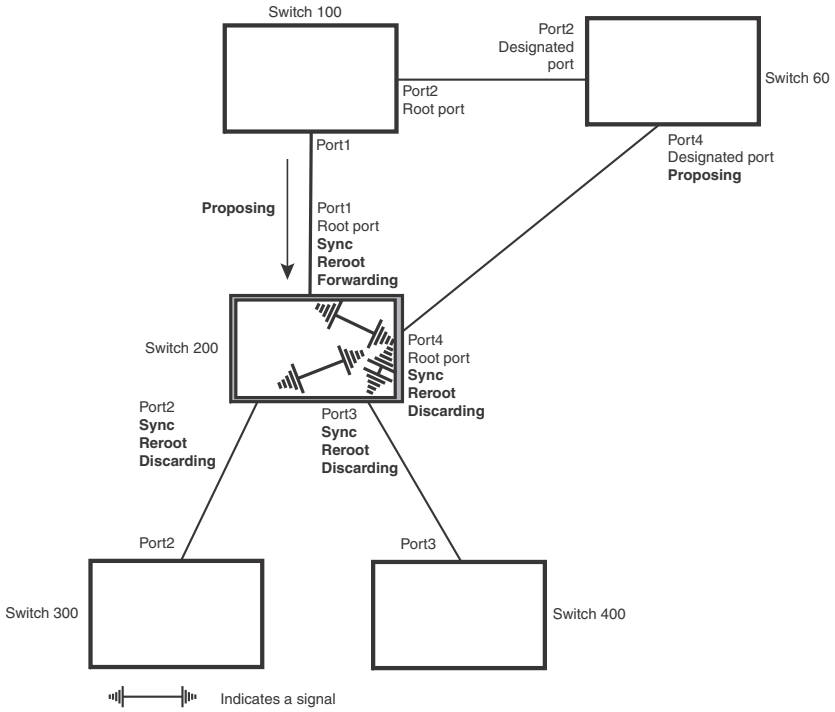
- **Proposing and Proposed** – The Designated port on the new root bridge (Port4/Switch 60) sends an RST BPDU that contains a proposing signal to Port4/Switch 200 to inform the port that it is ready to put itself in a forwarding state (Figure 39). RSTP algorithm determines that the RST BPDU that Port4/Switch 200 received is superior to what it can generate, so Port4/Switch 200 assumes a Root port role.

FIGURE 39 New root bridge sending a proposal flag



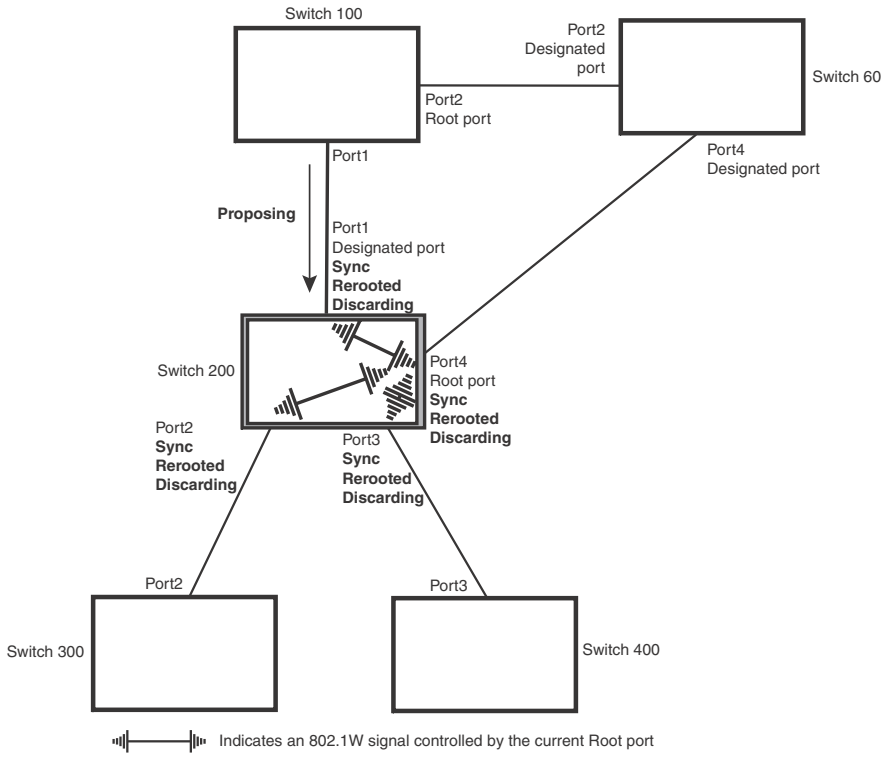
- **Sync and Reroot** – The Root port then asserts a sync and a reroot signal on all the ports on the bridge. The signal tells the ports that a new Root port has been assigned and they are to renegotiate their new roles and states. The other ports on the bridge assert their sync and reroot signals. Information about the old Root port is discarded from all ports. Designated ports change into discarding states (Figure 40).

FIGURE 40 Sync and reroot



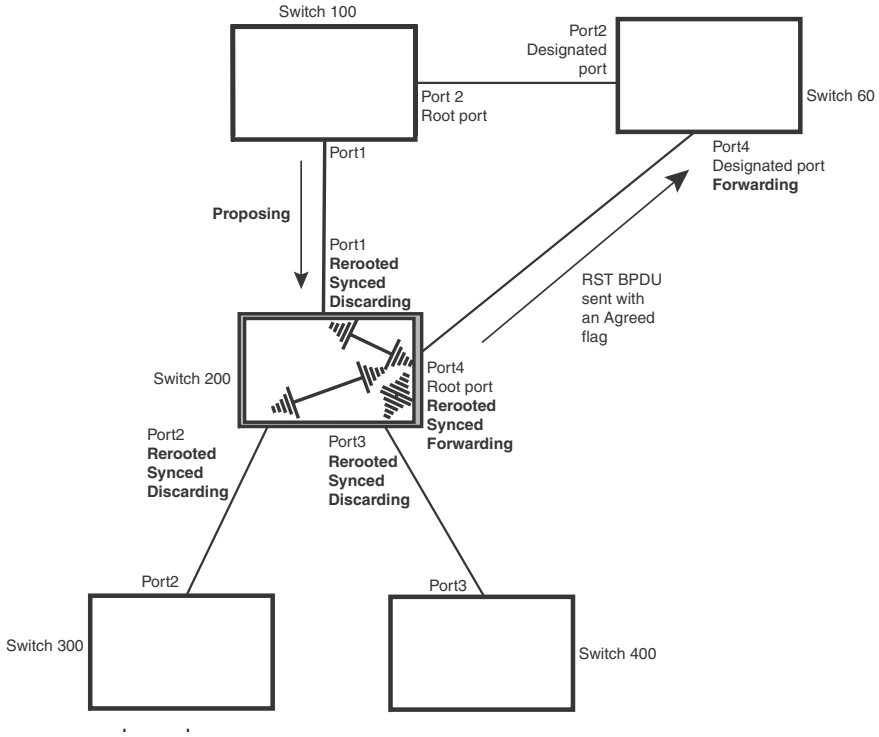
- **Sync and Rerooted** – When the ports on Switch 200 have completed the reroot phase, they assert their rerooted signals and continue to assert their sync signals as they continue in their discarding states. They also continue to negotiate their roles and states with their peer ports (Figure 41).

FIGURE 41 Sync and rerooted



- **Synced and Agree** – When all the ports on the bridge assert their synced signals, the new Root port asserts its own synced signal and sends an RST BPDU to Port4/Switch 60 that contains an agreed flag (Figure 41). The Root port also moves into a forwarding state.

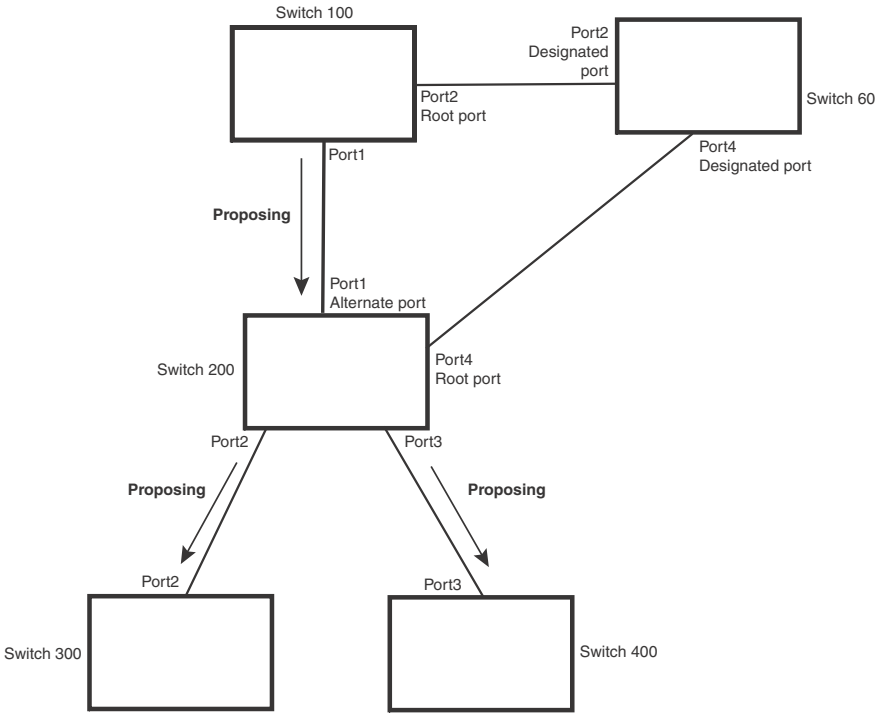
FIGURE 42 Rerooted, synced, and agreed



The old Root port on Switch 200 becomes an Alternate Port (Figure 43). Other ports on that bridge are elected to appropriate roles.

The Designated port on Switch 60 goes into a forwarding state once it receives the RST BPDU with the agreed flag.

FIGURE 43 Handshake completed after election of new root port



Recall that Switch 200 sent the agreed flag to Port4/Switch 60 and not to Port1/Switch 100 (the port that connects Switch 100 to Switch 200). Therefore, Port1/Switch 100 does not go into forwarding state instantly. It waits until two instances of the forward delay timer expires on the port before it goes into forwarding state.

At this point the handshake between the Switch 60 and Switch 200 is complete.

The remaining bridges (Switch 300 and Switch 400) may have to go through the reroot handshake if a new Root port needs to be assigned.

Convergence in a simple topology

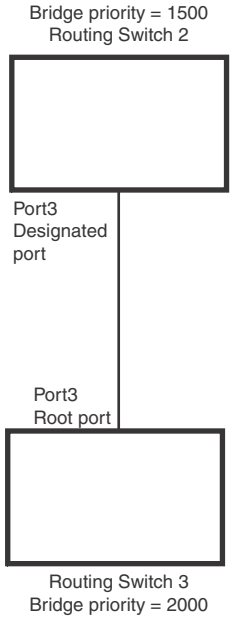
The examples in this section illustrate how RSTP convergence occurs in a simple Layer 2 topology at start-up.

NOTE
The remaining examples assume that the appropriate handshake mechanisms occur as port roles and states change.

Convergence at start up

In Figure 44, two bridges Switch 2 and Switch 3 are powered up. There are point-to-point connections between Port3/Switch 2 and Port3/Switch 3.

FIGURE 44 Convergence between two bridges



At power up, all ports on Switch 2 and Switch 3 assume Designated port roles and are at discarding states before they receive any RST BPDU.

Port3/Switch 2, with a Designated role, transmits an RST BPDU with a proposal flag to Port3/Switch 3. A ports with a Designated role sends the proposal flag in its RST BPDU when they are ready to move to a forwarding state.

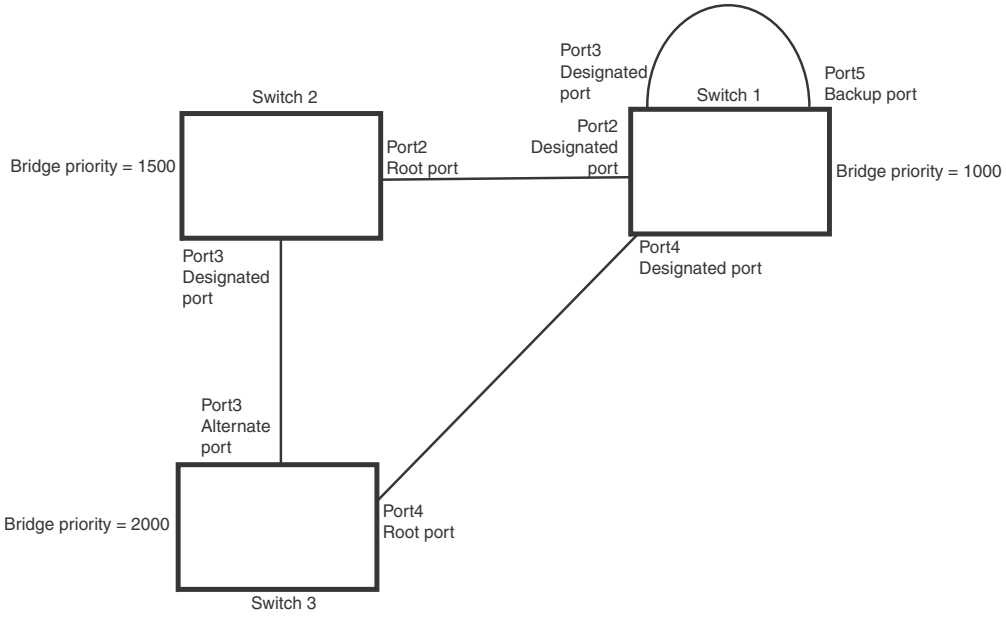
Port3/Switch 3, which starts with a role of Designated port, receives the RST BPDU and finds that it is superior to what it can transmit; therefore, Port3/Switch 3 assumes a new port role, that of a Root port. Port3/Switch 3 transmits an RST BPDU with an agreed flag back to Switch 2 and immediately goes into a forwarding state.

Port3/Switch 2 receives the RST BPDU from Port3/Switch 3 and immediately goes into a forwarding state.

Now RSTP has fully converged between the two bridges, with Port3/Switch 3 as an operational root port in forwarding state and Port3/Switch 2 as an operational Designated port in forwarding state.

Next, Switch 1 is powered up (Figure 45).

FIGURE 45 Simple Layer 2 topology



The point-to-point connections between the three bridges are as follows:

- Port2/Switch 1 and Port2/Switch 2
- Port4/Switch 1 and Port4/Switch 3
- Port3/Switch 2 and Port3/Switch 3

Ports 3 and 5 on Switch 1 are physically connected together.

At start up, the ports on Switch 1 assume Designated port roles, which are in discarding state. They begin sending RST BPDUs with proposal flags to move into a forwarding state.

When Port4/Switch 3 receives these RST BPDUs RSTP algorithm determines that they are better than the RST BPDUs that were previously received on Port3/Switch 3. Port4/Switch 3 is now selected as Root port. This new assignment signals Port3/Switch 3 to begin entering the discarding state and to assume an Alternate port role. As it goes through the transition, Port3/Switch 3 negotiates a new role and state with its peer port, Port3/Switch 2.

Port4/Switch 3 sends an RST BPDU with an agreed flag to Port4/Switch 1. Both ports go into forwarding states.

Port2/Switch 2 receives an RST BPDU. The RSTP algorithm determines that these RST BPDUs that are superior to any that any port on Switch 2 can transmit; therefore, Port2/Switch 2 assumes the role of a Root port.

The new Root port then signals all ports on the bridge to start synchronization. Since none of the ports are Edge ports, they all enter the discarding state and assume the role of Designated ports. Port3/Switch 2, which previously had a Designated role with a forwarding state, starts the discarding state. They also negotiate port roles and states with their peer ports. Port3/Switch 2 also sends an RST BPU to Port3/Switch 3 with a proposal flag to request permission go into a forwarding state.

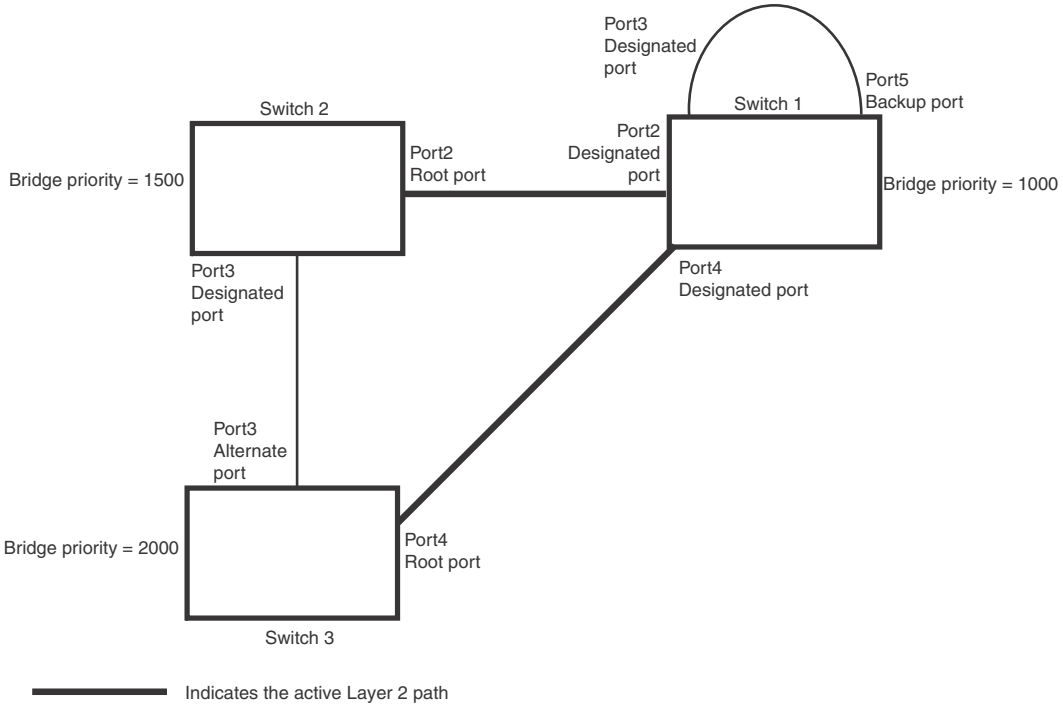
The Port2/Switch 2 bridge also sends an RST BPDU with an agreed flag Port2/Switch 1 that Port2 is the new Root port. Both ports go into forwarding states.

Now, Port3/Switch 3 is currently in a discarding state and is negotiating a port role. It received RST BPDUs from Port3/Switch 2. The RSTP algorithm determines that the RST BPDUs Port3/Switch 3 received are superior to those it can transmit; however, they are not superior to those that are currently being received by the current Root port (Port4). Therefore, Port3 retains the role of Alternate port.

Ports 3/Switch 1 and Port5/Switch 1 are physically connected. Port5/Switch 1 received RST BPDUs that are superior to those received on Port3/Switch 1; therefore, Port5/Switch 1 is given the Backup port role while Port3 is given the Designated port role. Port3/Switch 1, does not go directly into a forwarding state. It waits until the forward delay time expires twice on that port before it can proceed to the forwarding state.

Once convergence is achieved, the active Layer 2 forwarding path converges as shown in Figure 46.

FIGURE 46 Active Layer 2 path

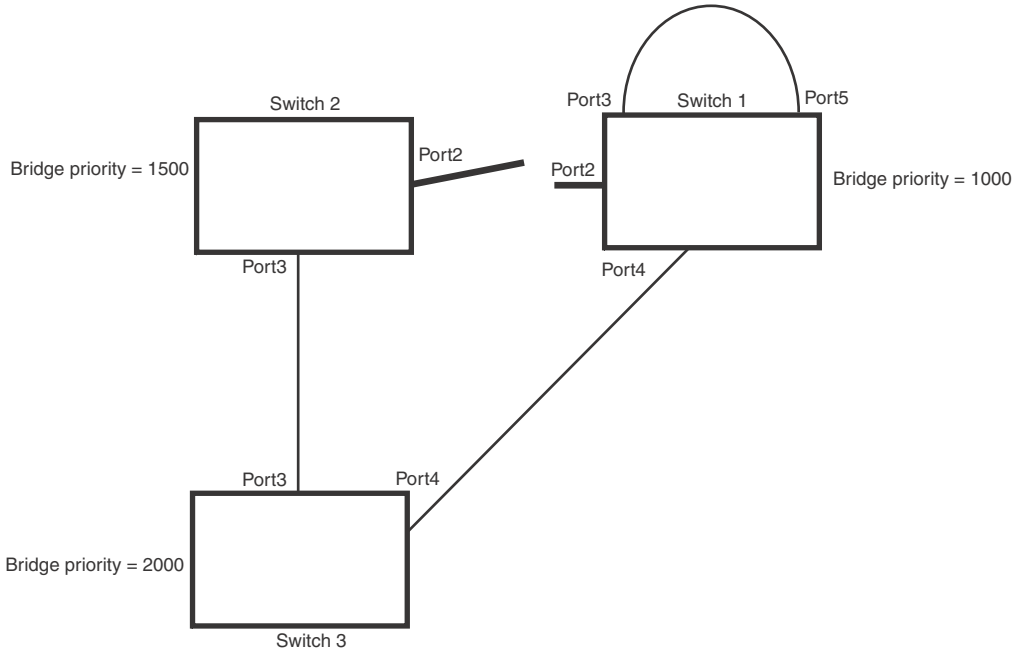


Convergence after a link failure

What happens if a link in the RSTP topology fails?

For example, Port2/Switch, which is the port that connects Switch 2 to the root bridge (Switch 1), fails. Both Switch 2 and Switch 1 notice the topology change (Figure 47).

FIGURE 47 Link failure in the topology



Switch 1 sets its Port2 into a discarding state.

At the same time, Switch 2 assumes the role of a root bridge since its root port failed and it has no operational Alternate port. Port3/Switch 2, which currently has a Designated port role, sends an RST BPDU to Switch 3. The RST BPDU contains a proposal flag and a bridge ID of Switch 2 as its root bridge ID.

When Port3/Switch 3 receives the RST BPDUs, RSTP algorithm determines that they are inferior to those that the port can transmit. Therefore, Port3/Switch 3 is given a new role, that of a Designated port. Port3/Switch 3 then sends an RST BPDU with a proposal flag to Switch 2, along with the new role information. However, the root bridge ID transmitted in the RST BPDU is still Switch 1.

When Port3/Switch 2 receives the RST BPDU, RSTP algorithm determines that it is superior to the RST BPDU that it can transmit; therefore, Port3/Switch 2 receives a new role; that of a Root port. Port3/Switch 2 then sends an RST BPDU with an agreed flag to Port3/Switch 3. Port3/Switch 2 goes into a forwarding state.

When Port3/Switch 3 receives the RST BPDU that Port3/Switch 2 sent, Port3/Switch 3 changes into a forwarding state, which then completes the full convergence of the topology.

Convergence at link restoration

When Port2/Switch 2 is restored, both Switch 2 and Switch 1 recognize the change. Port2/Switch 1 starts assuming the role of a Designated port and sends an RST BPDU containing a proposal flag to Port2/Switch 2.

When Port2/Switch 2 receives the RST BPDUs, RSTP algorithm determines that the RST BPDUs the port received are better than those received on Port3/Switch 3; therefore, Port2/Switch 2 is given the role of a Root port. All the ports on Switch 2 are informed that a new Root port has been assigned which then signals all the ports to synchronize their roles and states. Port3/Switch 2, which was the previous Root port, enters a discarding state and negotiates with other ports on the bridge to establish its new role and state, until it finally assumes the role of a Designated port.

Next, the following happens:

- Port3/Switch 2, the Designated port, sends an RST BPDU, with a proposal flag to Port3/Switch 3.
- Port2/Switch 2 also sends an RST BPDU with an agreed flag to Port2/Switch 1 and then places itself into a forwarding state.

When Port2/Switch 1 receives the RST BPDU with an agreed flag sent by Port2/Switch 2, it puts that port into a forwarding state. The topology is now fully converged.

When Port3/Switch 3 receives the RST BPDU that Port3/Switch 2 sent, RSTP algorithm determines that these RST BPDUs are superior to those that Port3/Switch 3 can transmit. Therefore, Port3/Switch 3 is given a new role, that of an Alternate port. Port3/Switch 3 immediately enters a discarding state.

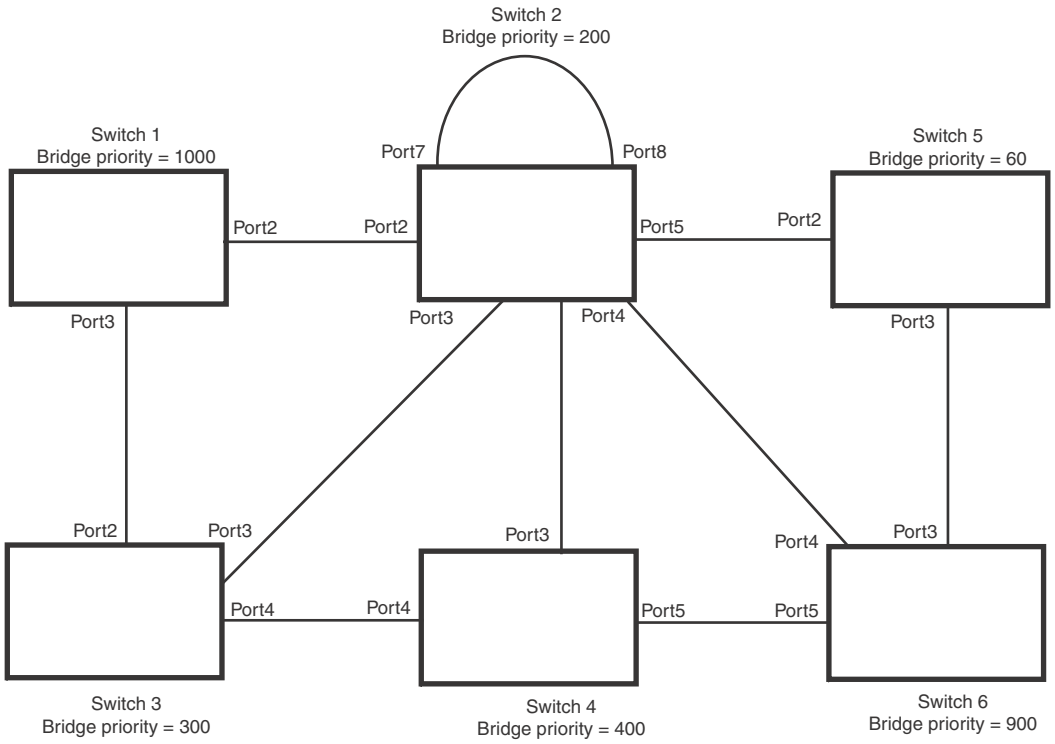
Now Port3/Switch 2 does not go into a forwarding state instantly like the Root port. It waits until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state. The wait, however, does not cause a denial of service, since the essential connectivity in the topology has already been established.

When fully restored, the topology is the same as that shown on Figure 45.

Convergence in a complex RSTP topology

The following is an example of a complex RSTP topology.

FIGURE 48 Complex RSTP topology



In Figure 48, Switch 5 is selected as the root bridge since it is the bridge with the highest priority. Lines in the figure show the point-to-point connection to the bridges in the topology.

Switch 5 sends an RST BPDU that contains a proposal flag to Port5/Switch 2. When handshakes are completed in Switch 5, Port5/Switch 2 is selected as the Root port on Switch 2. All other ports on Switch 2 are given Designated port role with discarding states.

Port5/Switch 2 then sends an RST BPDU with an agreed flag to Switch 5 to confirm that it is the new Root port and the port enters a forwarding state. Port7 and Port8 are informed of the identity of the new Root port. RSTP algorithm selects Port7 as the Designated port while Port8 becomes the Backup port.

Port3/Switch 5 sends an RST BPDU to Port3/Switch 6 with a proposal flag. When Port3/Switch 5 receives the RST BPDU, handshake mechanisms select Port3 as the Root port of Switch 6. All other ports are given a Designated port role with discarding states. Port3/Switch 6 then sends an RST BPDU with an agreed flag to Port3/Switch 5 to confirm that it is the Root port. The Root port then goes into a forwarding state.

Now, Port4/Switch 6 receives RST BPDUs that are superior to what it can transmit; therefore, it is given the Alternate port role. The port remains in discarding state.

Port5/Switch 6 receives RST BPDUs that are inferior to what it can transmit. The port is then given a Designated port role.

Next Switch 2 sends RST BPDUs with a proposal flag to Port3/Switch 4. Port3 becomes the Root port for the bridge; all other ports are given a Designated port role with discarding states. Port3/Switch 4 sends an RST BPDU with an agreed flag to Switch 2 to confirm that it is the new Root port. The port then goes into a forwarding state.

Now Port4/Switch 4 receives an RST BPDU that is superior to what it can transmit. The port is then given an Alternate port role, and remains in discarding state.

Likewise, Port5/Switch 4 receives an RST BPDU that is superior to what it can transmit. The port is also given an Alternate port role, and remains in discarding state.

Port2/Switch 2 transmits an RST BPDU with a proposal flag to Port2/Switch 1. Port2/Switch 1 becomes the Root port. All other ports on Switch 1 are given Designated port roles with discarding states.

Port2/Switch 1 sends an RST BPDU with an agreed flag to Port2/Switch 2 and Port2/Switch 1 goes into a forwarding state.

Port3/Switch 1 receives an RST BPDUs that is inferior to what it can transmit; therefore, the port retains its Designated port role and goes into forwarding state only after the forward delay timer expires twice on that port while it is still in a Designated role.

Port3/Switch 2 sends an RST BPDU to Port3/Switch 3 that contains a proposal flag. Port3/Switch 3 becomes the Root port, while all other ports on Switch 3 are given Designated port roles and go into discarding states. Port3/Switch 3 sends an RST BPDU with an agreed flag to Port3/Switch 2 and Port3/Switch 3 goes into a forwarding state.

Now, Port2/Switch 3 receives an RST BPDUs that is superior to what it can transmit so that port is given an Alternate port state.

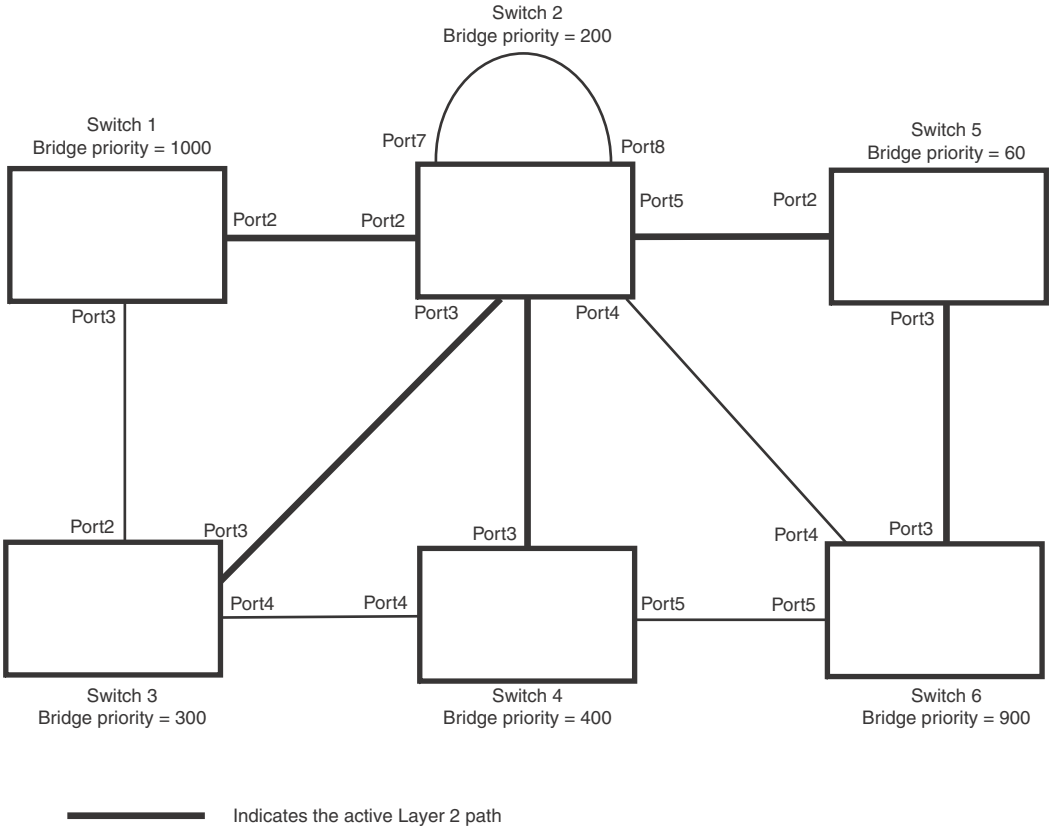
Port4/Switch 3 receives an RST BPDU that is inferior to what it can transmit; therefore, the port retains its Designated port role.

Ports on all the bridges in the topology with Designated port roles that received RST BPDUs with agreed flags go into forwarding states instantly. However, Designated ports that did not receive RST BPDUs with agreed flags must wait until the forward delay timer expires twice on those port. Only then will these port move into forwarding states.

The entire RSTP topology converges in less than 300 msec and the essential connectivity is established between the designated ports and their connected root ports.

After convergence is complete, Figure 49 shows the active Layer 2 path of the topology in Figure 48.

FIGURE 49 Active Layer 2 path in complex topology



Propagation of topology change

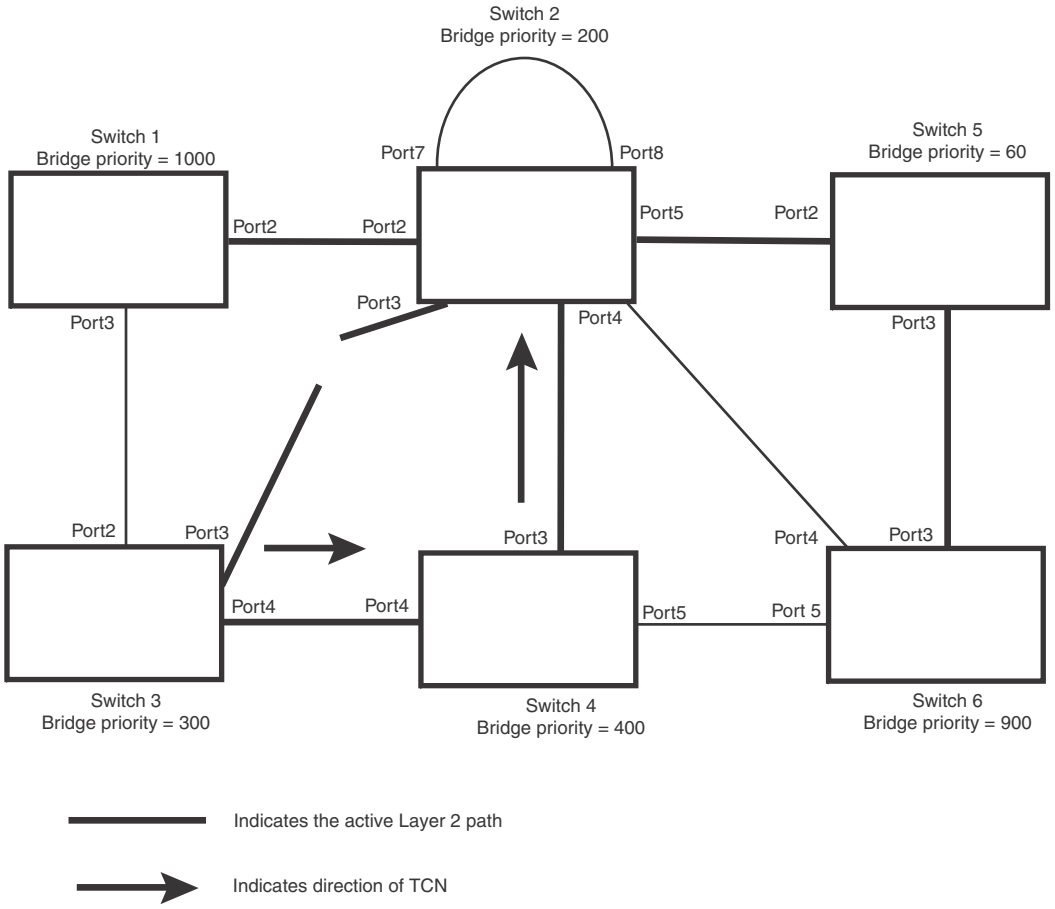
The Topology Change state machine generates and propagates the topology change notification messages on each port. When a Root port or a Designated port goes into a forwarding state, the Topology Change state machine on those ports send a topology change notice (TCN) to all the bridges in the topology to propagate the topology change.

NOTE
Edge ports, Alternate ports, or Backup ports do not need to propagate a topology change.

The TCN is sent in the RST BPDU that a port sends. Ports on other bridges in the topology then acknowledge the topology change once they receive the RST BPDU, and send the TCN to other bridges until all the bridges are informed of the topology change.

For example, Port3/Switch 2 in Figure 50, fails. Port4/Switch 3 becomes the new Root port. Port4/Switch 3 sends an RST BPDU with a TCN to Port4/Switch 4. To propagate the topology change, Port4/Switch 4 then starts a TCN timer on itself, on the bridge's Root port, and on other ports on that bridge with a Designated role. Then Port3/Switch 4 sends RST BPDU with the TCN to Port4/Switch 2. (Note the new active Layer 2 path in Figure 50.)

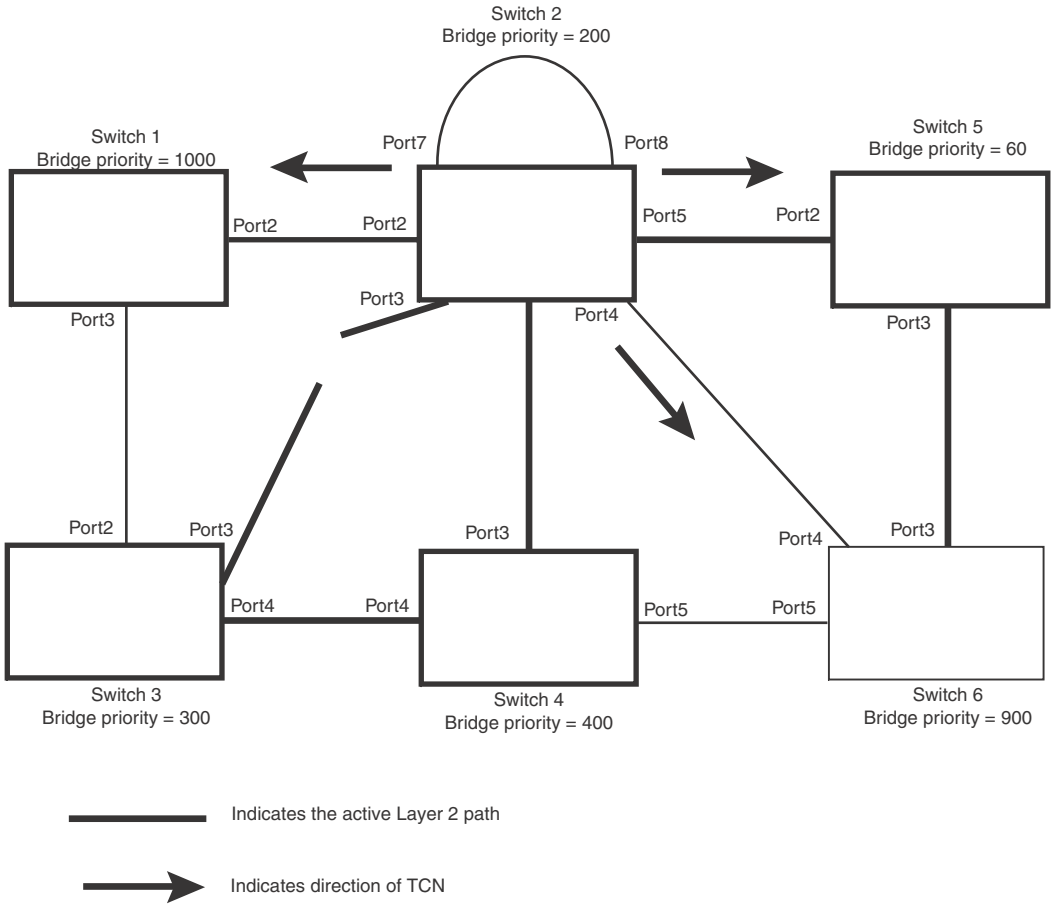
FIGURE 50 Beginning of topology change notice



Switch 2 then starts the TCN timer on the Designated ports and sends RST BPDUs that contain the TCN as follows (Figure 51):

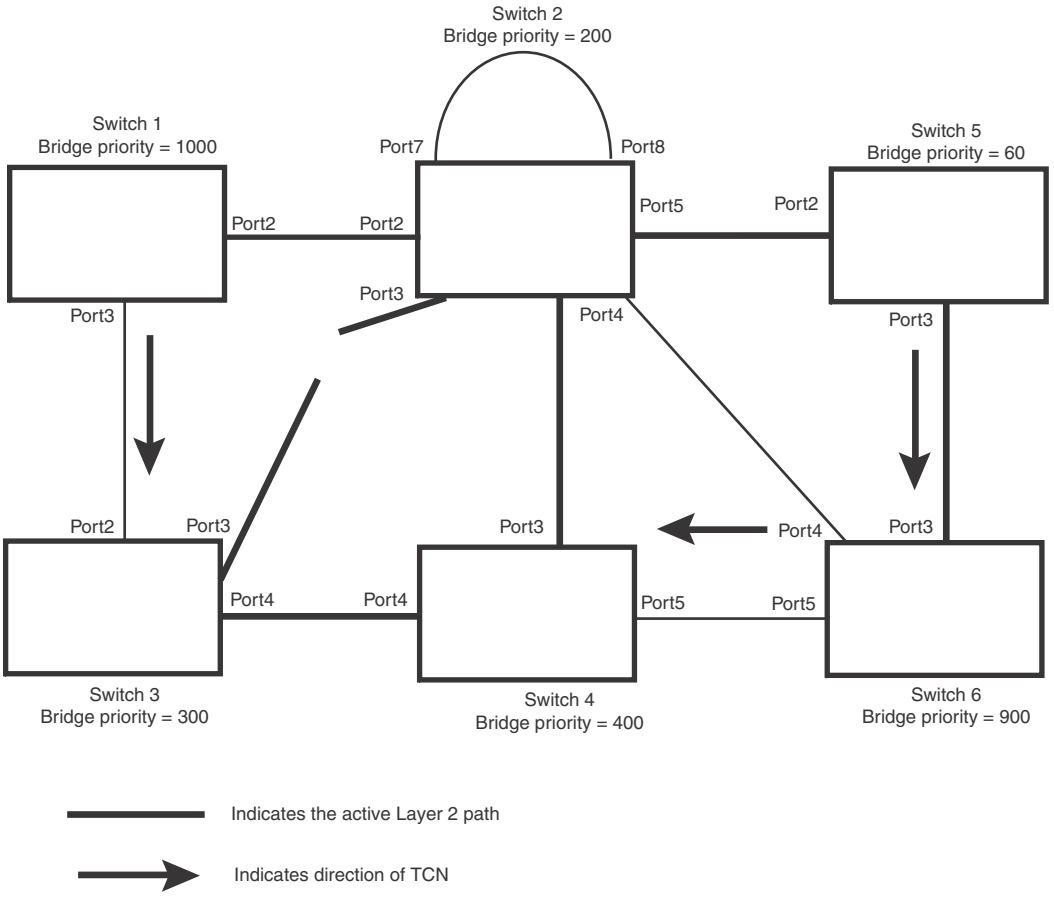
- Port5/Switch 2 sends the TCN to Port2/Switch 5
- Port4/Switch 2 sends the TCN to Port4/Switch 6
- Port2/Switch 2 sends the TCN to Port2/Switch 1

FIGURE 51 Sending TCN to bridges connected to Switch 2



Then FRY1, Switch 5, and Switch 6 send RST BPDUs that contain the TCN to Switch 3 and Switch 4 to complete the TCN propagation (Figure 52).

FIGURE 52 Completing the TCN propagation



Compatibility of RSTP with 802.1D

RSTP-enabled bridges are backward compatible with IEEE 802.1D bridges. This compatibility is managed on a per-port basis by the Port Migration state machine. **However, intermixing the two types of bridges in the network topology is not advisable if you want to take advantage of the rapid convergence feature.**

Compatibility with 802.1D means that an RSTP-enabled port can send BPDUs in the STP or 802.1D format when one of the following events occur:

- The port receives a legacy BPDU. A legacy BPDU is an STP BPDU or a BPDU in an 802.1D format. The port that receives the legacy BPDU automatically configures itself to behave like a legacy port. It sends and receives legacy BPDUs only.
- The entire bridge is configured to operate in an 802.1D mode when an administrator sets the
- bridge parameter to zero at the CLI, forcing all ports on the bridge to send legacy BPDUs only.

Once a port operates in the 802.1D mode, 802.1D convergence times are used and rapid convergence is not realized.

For example, in Figure 53, Switch 10 and Switch 30 receive legacy BPDUs from Switch 20. Ports on Switch 10 and Switch 30 begin sending BPDUs in STP format to allow them to operate transparently with Switch 20.

FIGURE 53 RSTP bridges with an 802.1D bridge

Once Switch 20 is removed from the LAN, Switch 10 and Switch 30 receive and transmit BPDUs in the STP format to and from each other. This state will continue until the administrator enables the **force-migration-check** command to force the bridge to send RSTP BPDU during a migrate time period. If ports on the bridges continue to hear only STP BPDUs after this migrate time period, those ports will return to sending STP BPDUs. However, when the ports receive RST BPDUs during the migrate time period, the ports begin sending RST BPDUs. The migrate time period is non-configurable. It has a value of three seconds.

NOTE
The IEEE standards state that RSTP bridges need to interoperate with 802.1D bridges. IEEE standards set the path cost of RSTP bridges to be between 1 and 200,000,000; whereas path cost of 802.1D bridges are set between 1 and 65,535. In order for the two bridge types to be able to interoperate in the same topology, the administrator needs to configure the bridge path cost appropriately. Path costs for either RSTP bridges or 802.1D bridges need to be changed; in most cases, path costs for RSTP bridges need to be changed.

Configuring RSTP parameters

The remaining RSTP sections explain how to configure the RSTP protocol on a PowerConnect.

You can enable or disable RSTP at the following levels:

- **Port-based VLAN** – Affects all ports within the specified port-based VLAN. When you enable or disable RSTP within a port-based VLAN, the setting overrides the global setting. Thus, you can enable RSTP for the ports within a port-based VLAN even when RSTP is globally disabled, or disable the ports within a port-based VLAN when RSTP is globally enabled.
- **Individual port** – Affects only the individual port. However, if you change the RSTP state of the primary port in a LAG group, the change affects all ports in the LAG group.

Enabling or disabling RSTP in a port-based VLAN

Use the following procedure to disable or enable RSTP on a PowerConnect on which you have configured a port-based VLAN. Changing the RSTP state in a VLAN affects only that VLAN.

To enable RSTP for all ports in a port-based VLAN, enter commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# rstp
```

Syntax: [no] rstp

Enabling or disabling RSTP on a single spanning tree

To globally enable RSTP for all ports of a single spanning tree, enter the following command.

```
NetIron(config)# rstp single
```

Syntax: [no] rstp single

Disabling or enabling RSTP on a port

The **rstp** command must be used to initially enable RSTP on ports. Both commands enable RSTP on all ports that belong to the VLAN or to the single spanning tree.

Once RSTP is enabled on a port, it can be disabled on individual ports. RSTP that have been disabled on individual ports can then be enabled as required.

NOTE
If you change the RSTP state of the primary port in a LAG group, the change affects all ports in that LAG group.

To disable or enable RSTP on a port, enter commands such as the following.

```
NetIron(config)# interface 1/1
NetIron(config-if-e1000-1/1)# no spanning-tree
```

Syntax: [no] spanning-tree

Changing RSTP bridge parameters

When you make changes to RSTP bridge parameters, the changes are applied to individual ports on the bridge.

To designate a priority for a bridge, enter a command such as the following at the VLAN level.

```
NetIron(config)# vlan 20
NetIron(config-vlan-20)# rstp priority 0
```

To make this change in the default VLAN, enter the following commands.

```
NetIron(config)# vlan 1
NetIron(config-vlan-1)# rstp priority 0
```

Syntax: [**rstp forward-delay** <*value*>] | [**hello-time** <*value*>] | [**max-age** <*time*>] | [**force-version** <*value*>] | [**priority** <*value*>]

The **forward-delay** <*value*> parameter specifies how long a port waits before it forwards an RST BPDU after a topology change. Possible values: 4 – 30 seconds. The default is 15 seconds.

The **hello-time** <*value*> parameter specifies the interval between two hello packets. Possible values: 1 - 10 seconds. The default is 2 seconds.

The **max-age** <*value*> parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. Possible values: 6 – 40 seconds. The default is 20 seconds.

The value of **max-age** must be greater than the value of **forward-delay** to ensure that the downstream bridges do not age out faster than the upstream bridges (those bridges that are closer to the root bridge).

The **force-version** <*value*> parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following values:

- 0 – The STP compatibility mode. Only STP (or legacy) BPDUs will be sent.
- 2 – The default. RST BPDUs will be sent unless a legacy bridge is detected. If a legacy bridge is detected, STP BPDUs will be sent instead.

The **priority** <*value*> parameter specifies the priority of the bridge. You can enter a value from 0 – 65535. A lower numerical value means a the bridge has a higher priority. Thus, the highest priority is 0. The default is 32768.

You can specify some or all of these parameters on the same command line.

Changing port parameters

The RSTP port commands can be enabled on individual ports or on multiple ports, such as all ports that belong to a VLAN.

The RSTP port parameters are preconfigured with default values. If the default parameters meet your network requirements, no other action is required.

You can change the following RSTP port parameters using the following methods.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# rstp ethernet 1/5 path-cost 15 priority 64
```

At the VLAN configuration level of the CLI:

Syntax: **rstp ethernet** <*slot*>/<*portnum*> **path-cost** <*value*> | **priority** <*value*> | **[admin-edge-port]** | **[admin-pt2pt-mac]** | **[force-migration-check]**

At the interface level of the CLI:

Syntax: **rstp [admin-edge-port]** | **[admin-pt2pt-mac]**

The **ethernet** <*slot*>/<*portnum*> parameter specifies the interface used.

The **path-cost** <*value*> parameter specifies the cost of the port's path to the root bridge. RSTP prefers the path with the lowest cost. You can specify a value from 1 – 20,000,000. Table 88 shows the recommended path cost values from the IEEE standards.

TABLE 88 Recommended path cost values of RSTP

Link speed	Recommended (default) RSTP path cost values	Recommended RSTP path cost range
Less than 100 kilobits per second	200,000,000	20,000,000 – 200,000,000
1 Megabit per second	20,000,000	2,000,000 – 200,000,000
10 Megabits per second	2,000,000	200,000 – 200,000,000
100 Megabits per second	200,000	20,000 – 200,000,000
1 Gigabit per second	20,000	2,000 – 200,000,000
10 Gigabits per second	2,000	200 – 20,000
100 Gigabits per second	200	20 – 2,000
1 Terabits per second	20	2 – 200
10 Terabits per second	2	1 – 20

The **priority** <*value*> parameter specifies the preference that RSTP gives to this port relative to other ports for forwarding traffic out of the topology. You can specify a value from 0 – 240, in increments of 16. If you enter a value that is not divisible by four, the software rounds to the nearest value that is divisible by four. The default is 128. A higher numerical value means a lower priority; thus, the highest priority is 8.

Set the **admin-edge-port** to enabled or disabled. If set to enabled, then the port becomes an edge port in the domain.

Set the **admin-pt2pt-mac** to enabled or disabled. If set to enabled, then a port is connected to another port through a point-to-point link. The point-to-point link increases the speed of convergence. This parameter, however, does not auto-detect whether or not the link is a physical point-to-point link.

The **force-migration-check** parameter forces the specified port to sent one RST BPDU. If only STP BPDUs are received in response to the sent RST BPDU, then the port will go return to sending STP BPDUs.

Displaying RSTP information

You can display a summary or details of the RSTP information.

To display a summary of RSTP, use the following command.

```
NetIron(config)#show rstp vlan 10
VLAN 10 - RSTP instance 0
------------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:

Bridge           Bridge Bridge Bridge Force    tx
Identifier       MaxAge Hello  FwdDly Version  Hold
hex              sec    sec    sec             cnt
0001000480a04000 20     2      15     Default  3

RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
0001000480a04000 0         0001000480a04000 Root  20  2   15

RSTP (IEEE 802.1w) Port Parameters:

      <--- Config Params -->|<-------------- Current state ----------------->
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/3   128 20000     T   F    DISABLED   DISABLED   0         0000000000000000
1/13  128 20000     T   F    DISABLED   DISABLED   0         0000000000000000
```

Syntax: show rstp [vlan <*vlan-id*>**]**

The **vlan** <*vlan-id*> parameter displays RSTP information for the specified port-based VLAN.

The **show RSTP display** command shows the information listed in Table 89.

TABLE 89 CLI display of RSTP summary

This field...	Displays...
VLAN ID	The port-based VLAN that owns the STP instance and the number of RSTP instances on that VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all RSTP information is for VLAN 1.
Bridge IEEE RSTP Parameters	
Bridge Identifier	The ID of the bridge.
Bridge Max Age	The configured max age for this bridge. The default is 20.
Bridge Hello	The configured hello time for this bridge.The default is 2.
Bridge FwdDly	The configured forward delay time for this bridge. The default is 15.
Force-Version	The configured force version value. One of the following value is displayed: • 0 – The bridge has been forced to operate in an STP compatibility mode. • 2 – The bridge has been forced to operate in an RSTP mode. (This is the default.)
txHoldCnt	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Root Bridge Parameters:	
Root Bridge Identifier	ID of the Root bridge that is associated with this bridge
Root Path Cost	The cost to reach the root bridge from this bridge. If the bridge is the root bridge, then this parameter shows a value of zero.
Designated Bridge Identifier	The bridge from where the root information was received.It can be from the root bridge itself, but it could also be from another bridge.
Root Port	The port on which the root information was received. This is the port that is connected to the Designated Bridge.
Max Age	The ***max age*** is derived from the Root port. An RSTP-enabled bridge uses this value, along with the hello and message age parameters to compute the effective age of an RST BPDU. The ***message age*** parameter is generated by the Designated port and transmitted in the RST BPDU. RST BPDUs transmitted by a Designated port of the root bridge contains a message value of zero. ***Effective age*** is the amount of time the Root port, Alternate port, or Backup port retains the information it received from its peer Designated port. Effective age is reset every time a port receives an RST BPDU from its Designated port. If a Root port does not receive an RST BPDU from its peer Designated port for a duration more than the effective age, the Root port ages out the existing information and recomputes the topology. If the port is operating in 802.1D compatible mode, then max age functionality is the same as in 802.1D (STP).
Hello	The hello value derived from the Root port. It is the number of seconds between two Hello packets.

TABLE 89 CLI display of RSTP summary (Continued)

This field...	Displays...
Fwd Dly	The number of seconds a non-edge Designated port waits until it can apply any of the following transitions, if the RST BPDU it receives does not have an agreed flag: • Discarding state to learning state • Learning state to forwarding state When a non-edge port receives the RST BPDU it goes into forwarding state within 4 seconds or after two hello timers expire on the port. Fwd Dly is also the number of seconds that a Root port waits for an RST BPDU with a proposal flag before it applies the state transitions listed above. If the port is operating in 802.1D compatible mode, then forward delay functionality is the same as in 802.1D (STP).
RSTP (IEEE 802.1W) Port Parameters	
Port Num	The port number shown in a slot#/port# format.
Pri	The configured priority of the port. The default is 128 or 0x80.
Port Path Cost	The configured path cost on a link connected to this port.
P2P Mac	Indicates if the point-to-point-mac parameter is configured to be a point-to-point link: • T – The link is configured as a point-to-point link. • F – The link is not configured as a point-to-point link. This is the default.
Edge port	Indicates if the port is configured as an operational Edge port: • T – The port is configured as an Edge port. • F – The port is not configured as an Edge port. This is the default.
Role	The current role of the port: • Root • Designated • Alternate • Backup • Disabled Refer to "Bridges and bridge port roles" on page 413 for definitions of the roles.
State	The port's current RSTP state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled Refer to "Bridge port states" on page 417 and "Edge port and non-Edge port states" on page 418.
Designated Cost	The best root path cost that this port received, including the best root path cost that it can transmit.
Designated Bridge	The ID of the bridge that sent the best RST BPDU that was received on this port.

To display detailed information about RSTP, using the following command.

```
NetIron(config)#show rstp detail
VLAN 10 - RSTP instance 0
------------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:

BridgeId 0001000480a04000, RootBridgeId 0001000480a04000
Control ports - ethernet 1/3 ethernet 1/13
ForceVersion 2, MigrateTime 3, TxHoldCount 3

RSTP (IEEE 802.1w) Port Parameters:

Port 1/3 - Role: DISABLED - State: DISABLED
Port 1/13 - Role: DISABLED - State: DISABLED
```

Syntax: show rstp detail [vlan <*vlan-id*>**]**

The **vlan** <*vlan-id*> parameter displays RSTP information for the specified port-based VLAN.

The **show RSTP detail** command shows the following information.

This field...	Displays...
VLAN ID	ID of the VLAN that owns the instance of RSTP and the number of RSTP instances on that VLAN.
Bridge ID	ID of the bridge.
Control ports	Ports assigned to the VLAN
forceVersion	the configured version of the bridge: • 0 – The bridge has been forced to operate in an STP compatible mode. • 2 – The bridge has been forced to operate in an RSTP mode.
MigrateTime	The number of seconds the bridge took to migrate from STP to RSTP mode.
txHoldCount	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Port	ID of the port in slot#/port# format.
Role	The current role of the port: • Root • Designated • Alternate • Backup • Disabled Refer to "Bridges and bridge port roles" on page 413 for definitions of the roles.
State	The port's current RSTP state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled Refer to "Bridge port states" on page 417 and "Edge port and non-Edge port states" on page 418.

Configuring RSTP under an ESI VLAN

RSTP can also be configured under a VLAN that is part of a user-configured ESI. For example, to enable RSTP on a VLAN that is part of an ESI, configure the following commands.

```
NetIron(config)# esi customer1 encapsulation cvlan
NetIron(config-esi-customer1)# vlan 100
NetIron(config-esi-customer1-vlan-100)# rstp
```

Metro Ring Protocol 13

Overview

The following Metro Ring Protocol features are supported by the NetIron MLX Series device.

- MRP Phase 1
- MRP Phase 2
- Foundry MRP Alarm RHP
- Foundry MRP Diagnostics

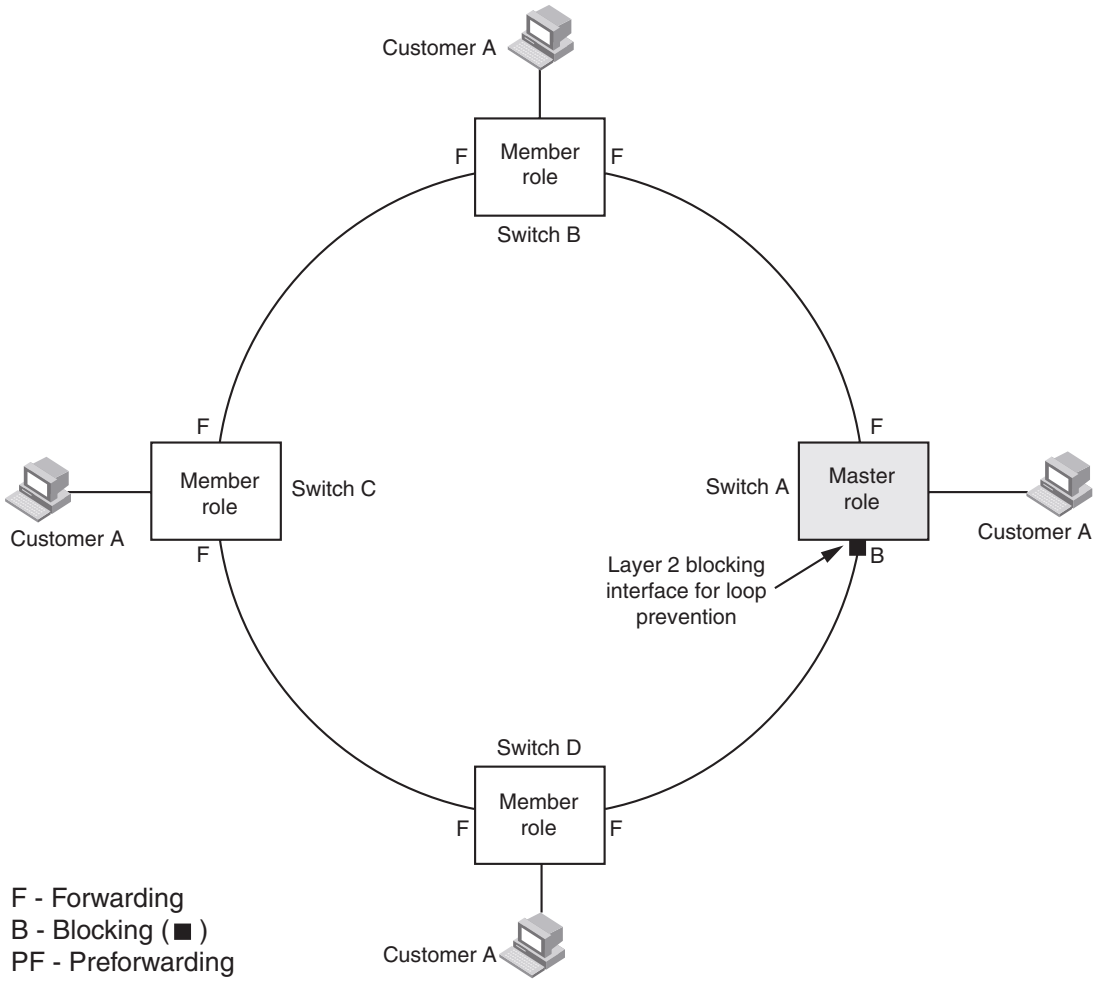
Metro Ring Protocol (MRP)

The MRP is a proprietary protocol that prevents layer 2 loops and provides fast reconvergence in ring topologies. It is an alternative to Spanning Tree Protocol (STP) and is especially useful in Metropolitan Area Networks (MANs) where using 802.1D STP has the following drawbacks:

- 802.1D recommends a maximum bridge diameter of seven nodes with standard timers. MRP is capable of many more nodes than this.
- 802.1D has a slow reconvergence time, taking many seconds or even minutes. MRP can detect and heal a break in the ring in under one second.

Figure 54 shows a simple metro ring.

FIGURE 54 MRP – normal state



The ring in this example consists of four Dell switch nodes that support MRP. Each node has two ring interfaces and the interfaces are all in one port-based vlan. There are customer networks utilising the nodes and layer 2 traffic is forwarded to and from the customer networks through the ring. Each customer interface can be in the same vlan as the ring or in a separate vlan under control of MRP as part of a topology group.

For each discrete ring one node is configured in the master role for the MRP ring. One of the two ring interfaces on the master node is configured as the primary interface, the other is the secondary interface. The primary interface originates Ring Health Packets (RHPs) which are used to monitor the health of the ring. An RHP is forwarded on the ring to the next interface until it reaches the secondary interface of the master node. On receipt of an RHP the secondary interface transitions into blocking mode to prevent a layer 2 loop.

The following MRP features are supported by Brocade NetIron XMR Series devices.

- MRP Phase 1
- MRP Phase 2
- MRP Alarm RHP

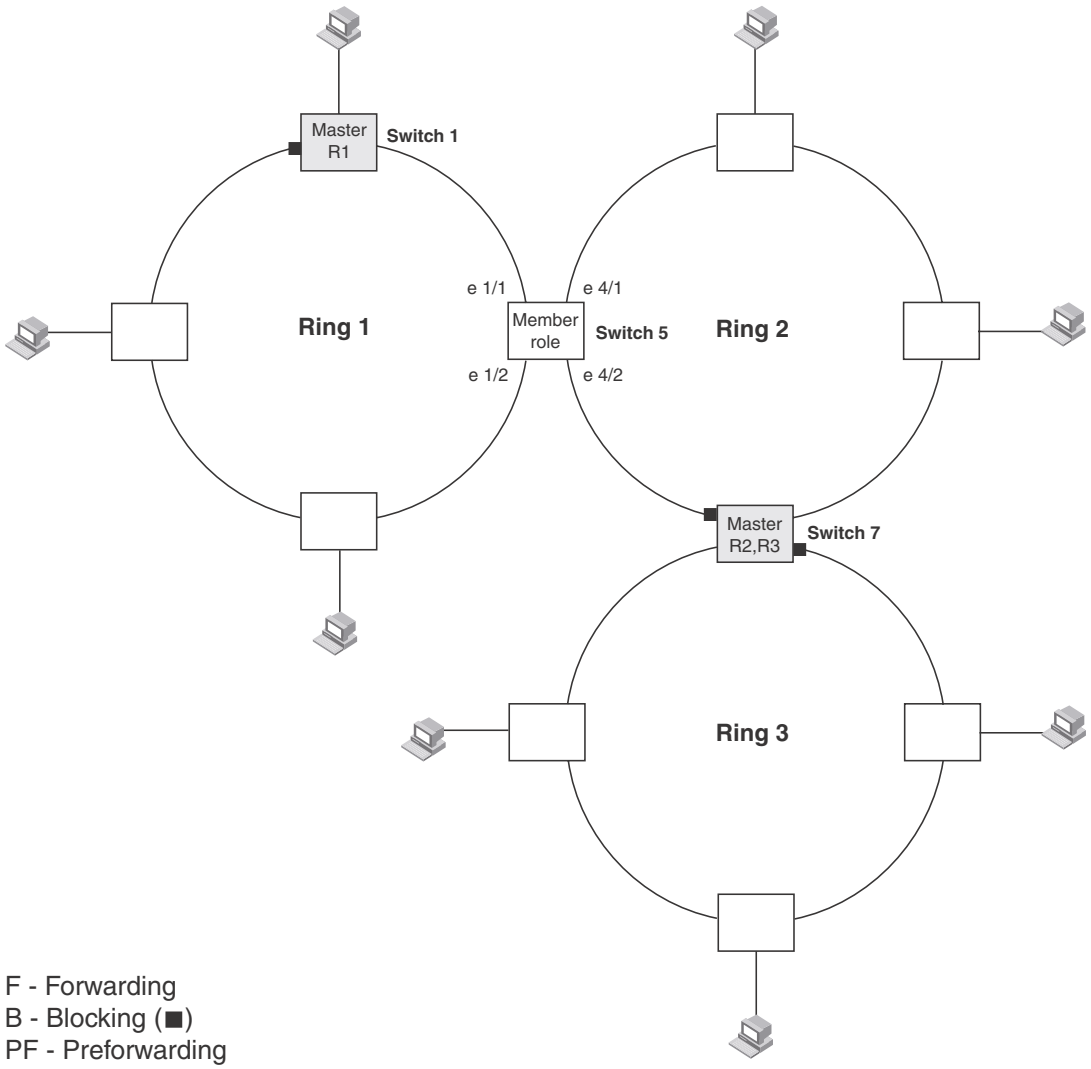
- MRP Diagnostics

NOTE
When you configure MRP, it is recommended that you disable the secondary ring interface on the master node before beginning or changing the ring configuration. Disabling an interface prevents a layer 2 loop from occurring while you are configuring MRP on the ring nodes. Once you have completed the MRP configuration and enabled it on all the nodes, you should re-enable the secondary ring interface.

MRP rings without shared interfaces (MRP Phase 1)

MRP Phase 1 allows you to configure multiple MRP rings, as shown in Figure 55, but the rings cannot share the same interfaces. For example, you cannot configure ring 1 and ring 2 to share interfaces ethernet 1/1 and 1/2 on switch 5. Each ring must remain an independent ring and RHP packets are processed within each ring.

FIGURE 55 MRP – multiple rings, no shared interfaces

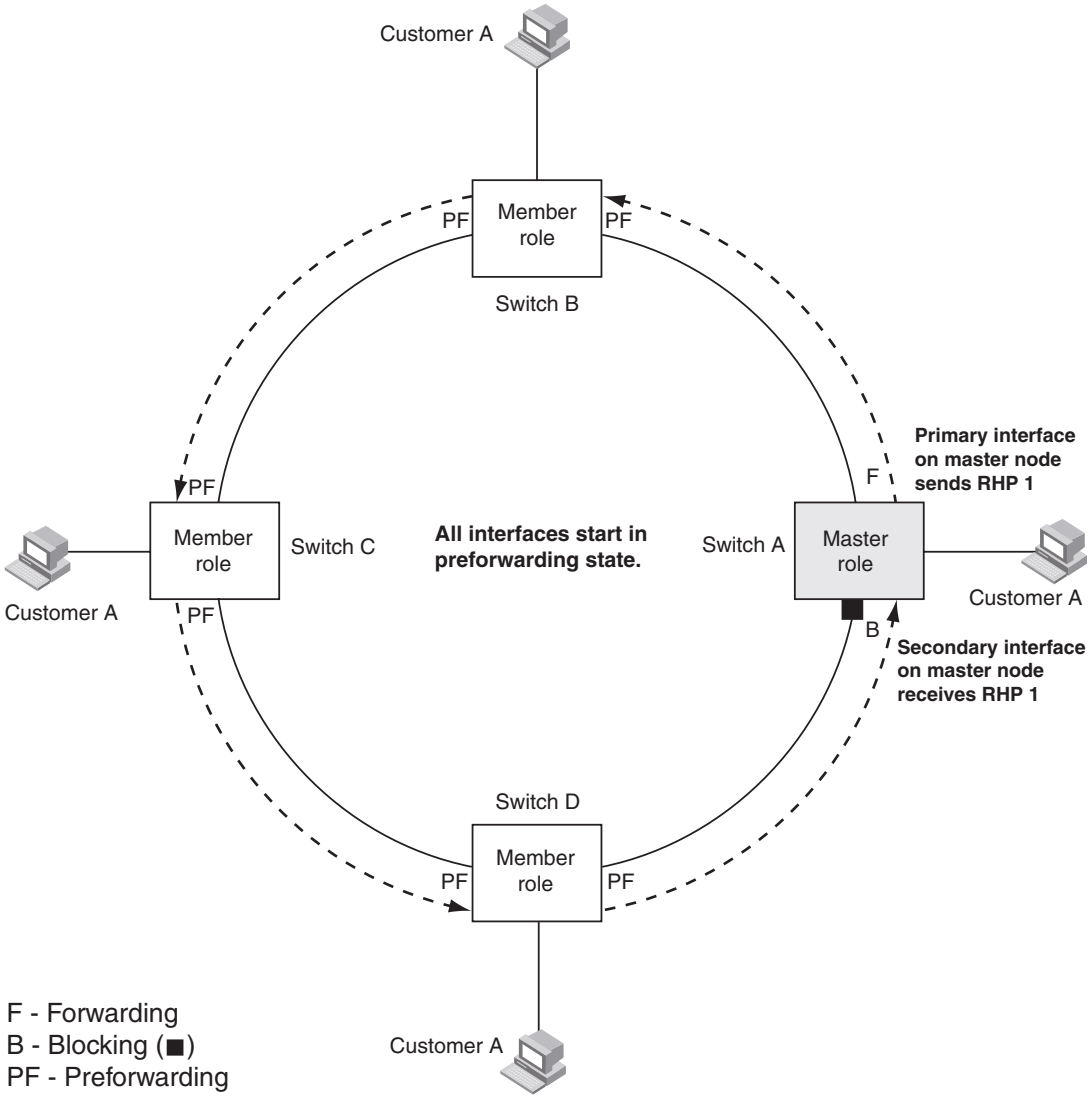


In the above example, switch 5 and switch 7 are configured with multiple MRP rings however each ring has discrete ring interfaces allocated to it to prevent any sharing. Any ring node can be the master for its ring and a node can be the master for more than one ring as shown on switch 7 due to the separation of rings.

Ring initialization

Figure 56 shows the initial state of the ring, when MRP is first enabled on the ring's switches. On the master the primary interface starts in forwarding mode and the secondary interface starts in blocking mode. All ring interfaces on member nodes begin in the preforwarding state (PF).

FIGURE 56 MRP ring – initial state



An RHP is an MRP protocol packet used to monitor the health of the ring. The source address is the MAC address of the master node and the destination MAC address is a protocol address for MRP. The Master node generates RHPs and sends them on the ring. The state of a ring interface is influenced by the RHPs.

A ring interface can have one of the following MRP states:

- **Preforwarding (PF)** – The interface will forward RHPs and learn MAC addresses but won't forward data for the ring. All ring interfaces start in this state when you enable MRP except the master node. A blocking interface transitions to preforwarding when the preforwarding timer expires and no RHP's have been received.
- **Forwarding (F)** – The interface will forward RHP's and data for the ring. On member switches an interface transitions from preforwarding to forwarding when the preforwarding time expires or the interface receives an RHP with the forwarding bit set. A break in the ring is indicated if the secondary interface on the master fails to receive an RHP within the preforwarding timer and the interface transitions from blocking to forwarding to heal the ring.

- **Blocking (B)** – The interface can process RHPs, but cannot forward data for the ring. Only the secondary interface on the master node can be blocking.

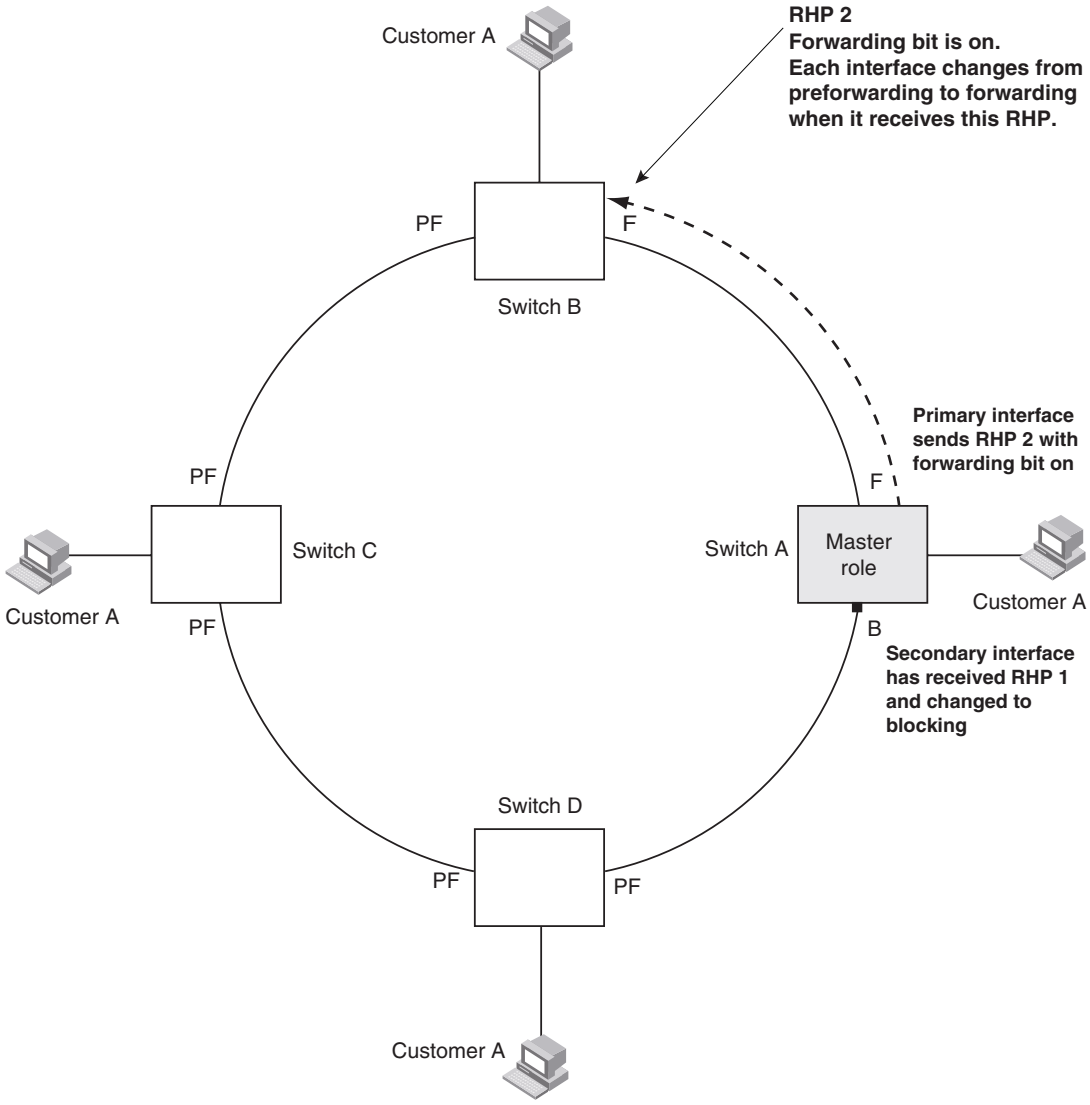
NOTE
The configured preforwarding time defines the number of milliseconds the interface will remain in a state before changing to the next state without receiving an RHP.

When MRP is enabled, all interfaces begin in the preforwarding state and the primary interface on the master node immediately sends an RHP (RHP 1 in Figure 56) onto the ring. The secondary interface on the master node listens for the RHP:

- If the secondary interface receives the RHP, all links in the ring are up and the interface changes its state to blocking. The primary interface then sends another RHP (RHP 2 in Figure 57) with its forwarding bit set on. As each of the member interfaces receives the RHP, the interfaces change their state to forwarding. Typically, this occurs in sub-second time. The ring very quickly enters the fully initialized state.
- If the secondary interface does not receive the RHP by the time the preforwarding time expires, a break has occurred in the ring. The secondary interface changes its state to forwarding. The ring is not intact, but data is still forwarded among the nodes using the links that are up.

Figure 57 shows an example.

FIGURE 57 MRP ring – from preforwarding to forwarding

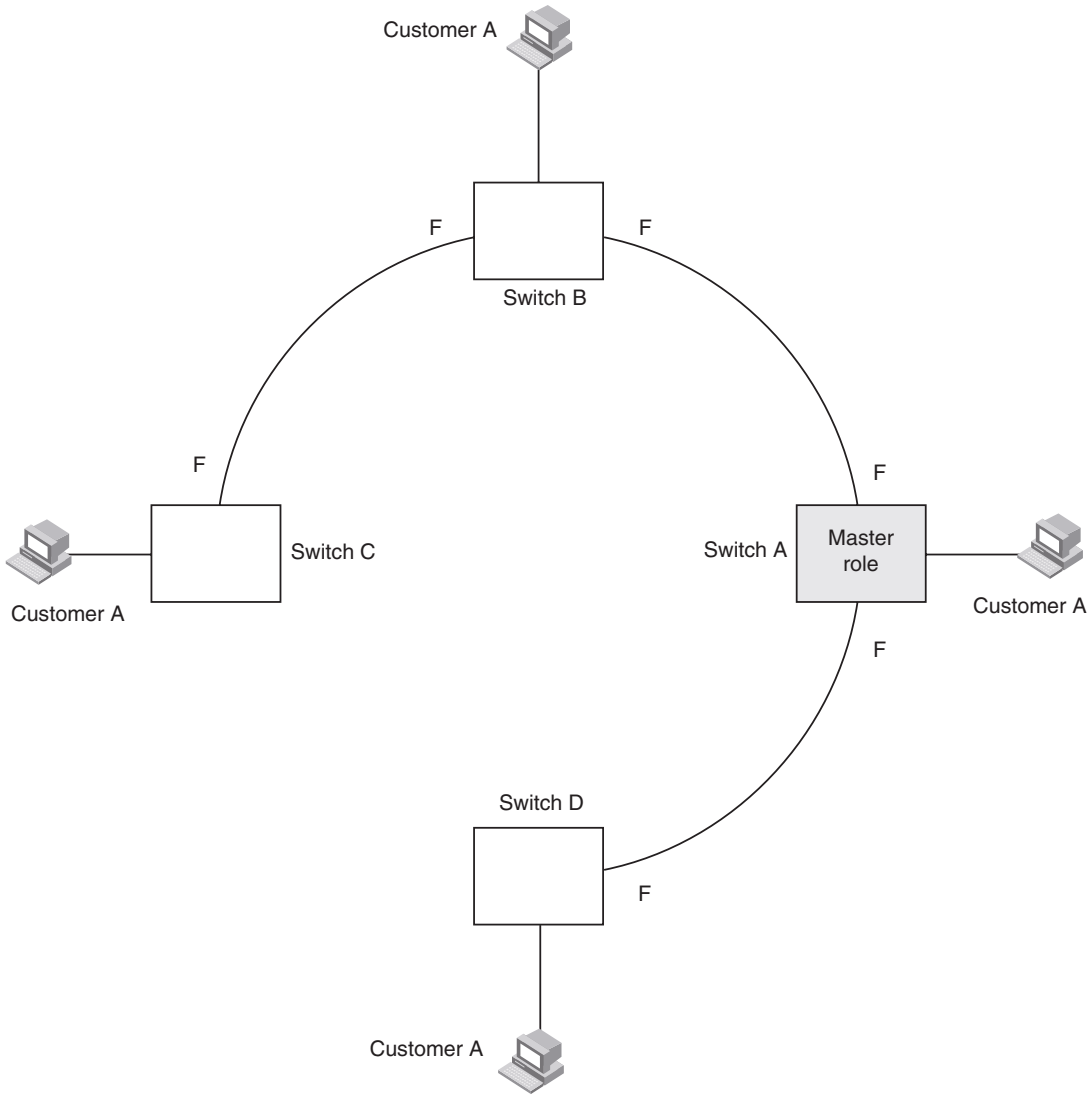


Each RHP also has a sequence number. MRP can use the sequence number to determine the round-trip time for RHPs in the ring. Refer to "Using MRP diagnostics" on page 479.

How ring breaks are detected and healed

Figure 58 shows ring interface states following a link break. MRP quickly heals the ring and preserves connectivity among the customer networks.

FIGURE 58 MRP ring – ring break



If a break in the ring occurs, MRP heals the ring by changing the states of some of the ring interfaces:

- **Blocking interface** – When the secondary interface on the master node transitions to a blocking state it sets a timer defined by the preforwarding time configured. If the timer expires before the interface receives a ring RHP, the interface changes state to preforwarding. Once the secondary interface state is preforwarding:
  - If the interface receives an RHP, the interface changes back to the blocking state and resets the timer.
  - If the interface does not receive an RHP for its ring before the preforwarding time expires, the interface changes to the forwarding state, as shown in Figure 58.
- **Forwarding interfaces** – All member interfaces remain in the forwarding state unless the physical interface is in an error condition.

When the link is repaired, the associated MRP interfaces come up in the preforwarding state allowing RHPs to be forwarded around the ring and finally reach the secondary interface on the master node:

- If an RHP reaches the master node's secondary interface, the ring is intact, the secondary interface changes to blocking. The master node sets the forwarding bit on in the next RHP. When the restored interfaces receive this RHP, they immediately change state to forwarding.
- If an RHP does not reach the master node's secondary interface, the ring is still broken. The master node does not send an RHP with the forwarding bit on. In this case, the restored interfaces remain in the preforwarding state until the preforwarding timer expires, then change to the forwarding state.

MRP alarm RHP enhancement

Prior to the enhancement detection of ring breaks were completely timer based. If the ring master fails to receive RHPs for a period of 3 "hello times" (by default the hello time is 100 ms) this indicates that the ring is broken in some manner. This initiates a topology change as described in the previous section. The convergence time associated with such an event could take several hundred milliseconds.

This enhancement enables ring nodes to rapidly notify the master of link failures. To understand the mechanism we introduce the concept of downstream switches in the ring and how member switches determine the primary and secondary ring interfaces. Remember that a primary ring interface sends RHPs and a secondary ring interface receives RHPs.

To fully understand the mechanism the reader needs to be aware of the concept of shared interfaces and interface owner ID's which are a function of MRP phase 2.

A downstream switch is defined as the next switch that will receive the ring RHP originated from the master primary interface for a particular ring. In Figure 59 switch B is downstream from the master, switch C is downstream from switch B and so on and so forth. In addition it should be noted that a member switch identifies which ring interface is secondary for each discrete ring by virtue of the receipt of RHPs for that ring. In a topology with shared interfaces a single physical interface can therefore be a primary ring interface for one ring and a secondary ring interface for another ring. It should be noted that the output of the 'show metro' command as well as the configuration will change if the primary and secondary ring interfaces of the master are swapped. This keeps the identification of interface roles consistent with the flow of RHPs for discrete ring instances.

When a link is detected to be down on a member switch secondary ring interface due to a link failure an alarm RHP, which is an RHP with the alarm bit set, is sent from the primary ring interface towards the ring master, notifying the master of the failure.

The destination MAC address in the alarm is the ring MAC address. The MAC address will be in the format 0304.8000.00xx where 'xx' is the ring number in hexadecimal.

For example ring 100 = 0304.8000.0064. This ensures that the packet is hardware forwarded all the way to the master. When the master in the ring receives this alarm the secondary interface is immediately transitioned from blocking to forwarding.

NOTE
In the event of a shared interface failing the alarm RHP packet is only sent by the owner ring of the failed interface. If all rings configured on a shared interface were to generate alarms then the respective master switches for each ring would start forwarding on both interfaces creating a loop condition. By restricting alarm generation to the owner ring we ensure that only one master switch is notified to ensure that the ring heals. The owner ring ID should be the highest priority ring configured on the shared interface.

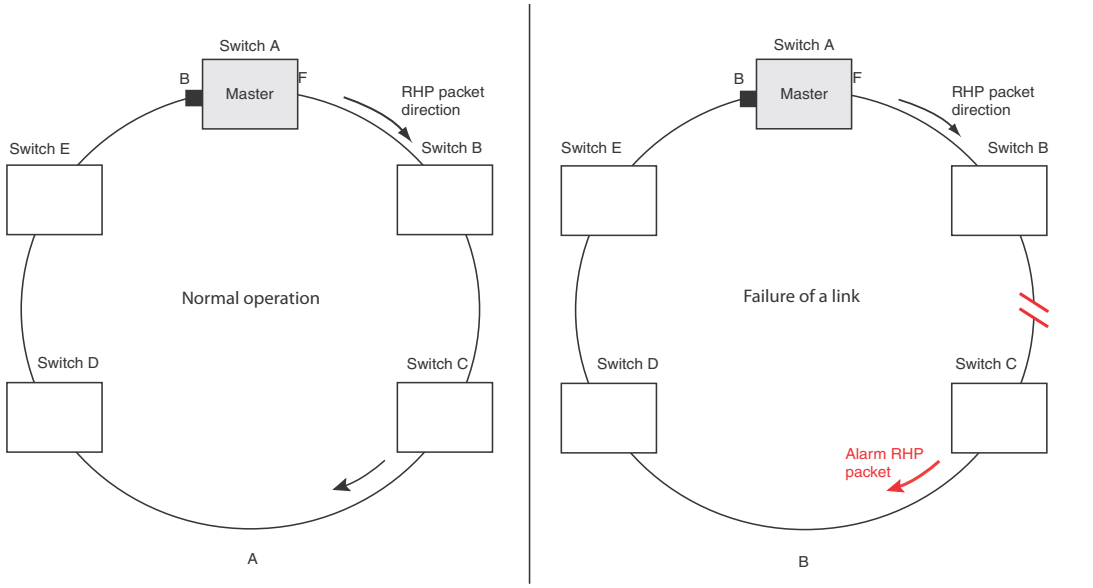
Operation of the alarm RHP enhancement is shown in Figure 59 and described below:

When the link between switch B and switch C fails, the downstream switch detects the failure of the link associated with its secondary ring interface and generates an alarm. The following is the complete sequence of events that occurs.

1. The downstream switch C detects a link down event on the link to its upstream neighbour switch B.
2. Switch C sends a single RHP packet with the alarm bit set. The RHP packet is sent in the same direction of flow as that of the normal RHP packets.
3. Switch A receives the alarm on the secondary ring interface that was sent by switch C. It is now aware that the ring is broken even though the preforwarding timer for blocking to preforwarding may not have expired.
4. Switch A immediately transitions its secondary interface from blocking to forwarding to heal the ring.
5. RHP packets continue to be sent on the primary interface by switch A to detect when the ring has been healed.

From a user perspective there is no other difference in the behavior of the ring other than the rapid convergence due to link failures. There is no CLI command required to enable this feature.

FIGURE 59 An MRP ring under normal operation (A) and after detection of a failure in the ring (B)

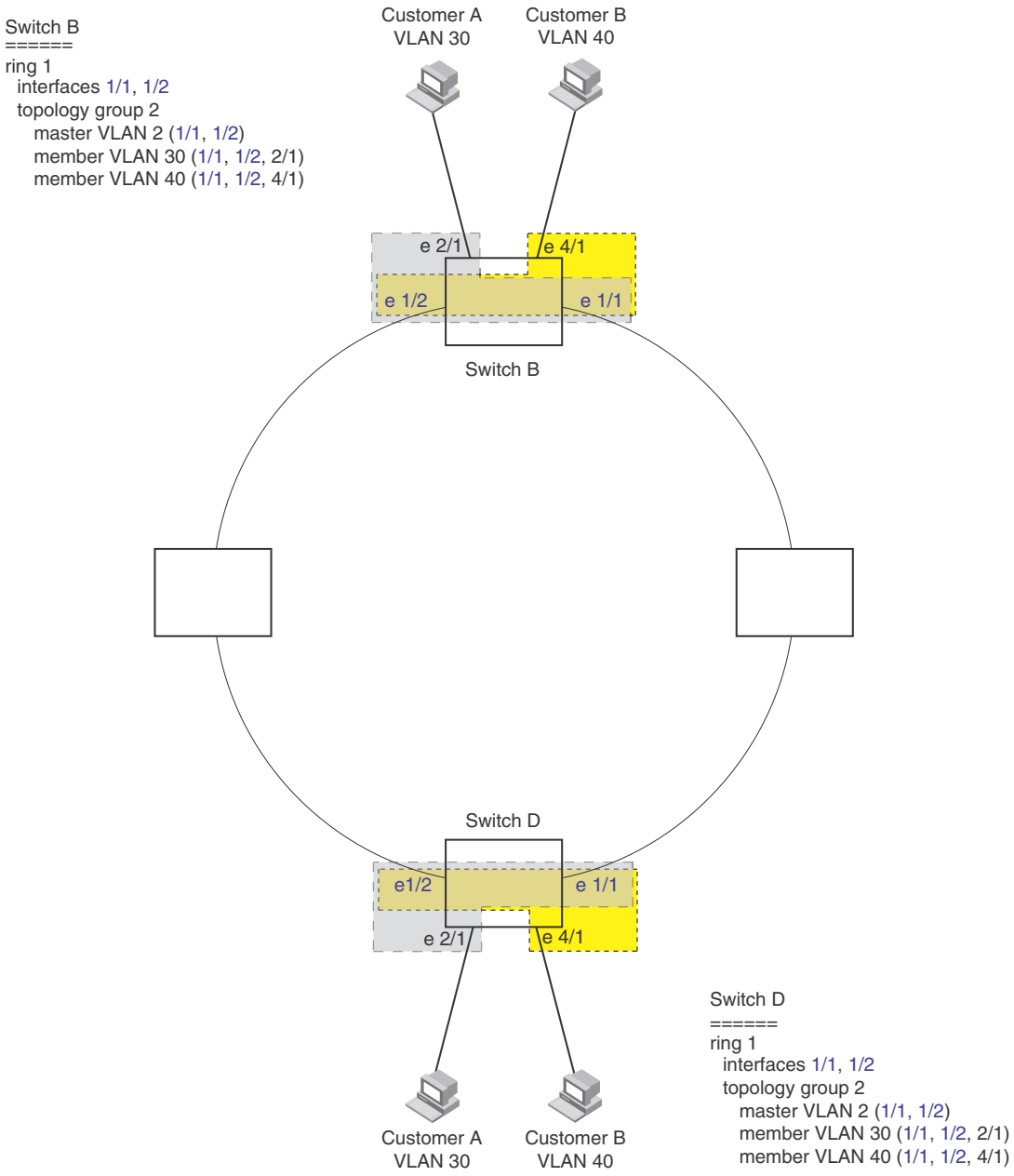


Master VLANs and member VLANs in a topology group

The reader is referred to chapter 16, "Topology Groups"for further information on topology group concepts and operation.

All the ring interfaces must be placed into the master vlan for the topology group. Customers configured with member vlans inherit the configuration of the topology group master vlan and have equivalent layer 2 connectivity across the ring. Figure 60 shows an example.

FIGURE 60 MRP ring – ring vlan and customer vlan



In this example each customer has their own vlan. Customer A has vlan 30 and customer B has vlan 40.

Customer A host attached to switch D on an interface in vlan 30 can reach the customer A host attached to switch B on an interface in vlan 30 through the ring at layer 2. The same mechanism is used to connect customer B hosts on vlan 40.

Customer A and customer B traffic is separated by using different vlans.

You can configure MRP separately on each customer vlan. However, this is impractical if you have many customers. To simplify configuration when you have a lot of customers (and therefore a lot of vlans), you can use a topology group.

A topology group enables you to control forwarding in multiple vlans using a single instance of a layer 2 protocol such as MRP. A topology group contains a master vlan and member vlans. The master vlan contains all the configuration parameters for the layer 2 protocol (STP, MRP, or VSRP). The member vlans use the layer 2 configuration of the master vlan.

In Figure 60, vlan 2 is the master vlan and contains the MRP configuration parameters for ring 1. vlan 30 and vlan 40, the customer vlans, are member vlans in the topology group. Since a topology group is used, a single instance of MRP provides redundancy and loop prevention for both the customer vlans.

If you use a topology group:

- The master vlan must contain the ring interfaces.
- The ring interfaces must be tagged as they will be used for multiple vlans.
- The member vlan for a customer must contain the two ring interfaces and the interfaces for the customer.

Refer to "MRP CLI example" on page 482 for the configuration commands required to implement the MRP configuration shown in Figure 60.

Configuring MRP

To configure MRP, perform the following tasks for each discrete ring:

- On the switch identified as the ring master disable the secondary ring interface. This manually prevents a layer 2 loop from occurring during configuration.
- Configure each switch for MRP one at a time following the planned flow of RHP's
- On each switch in the path add an MRP ring to a port-based vlan. When you add a ring, the CLI changes to the configuration level for the ring, where you can do the following:
  - On the master node configure the master ring role.
  - Specify the two MRP interfaces for the ring
  - Option: Specify a name for the ring. Dell recommends that you have a naming convention for your MRP rings and consistently apply names for all the rings in the topology.
  - Option: Change the hello time and the preforwarding time. These parameters control how quickly failover occurs if the master fails to receive RHPs for the ring.
  - Enable the ring.
- Re-enable the interface you disabled in step one. MRP will prevent loops when enabled on all devices in the ring.

When using topology groups the ring configuration must be added to the master-vlan for the group. For further information refer to 16, "Topology Groups".

Adding an MRP ring to a vlan

NOTE
If you plan to use a topology group make sure you configure MRP on the topology group's master vlan.

To add a MRP ring to a vlan, enter commands such as the following.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# metro-ring 1
NetIron(config-vlan-2-mrp-1)# name CustomerA
NetIron(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
NetIron(config-vlan-2-mrp-1)# enable
```

These commands configure an MRP ring in vlan 2 with a ring ID of 1, a ring name of CustomerA. If the node is the master then the following command is used to specify the node as the master for the ring.

NetIron(config-vlan-2-mrp-1)# master

The ring interfaces are 1/1 and 1/2. The first interface listed will be allocated as the primary interface and the second will be allocated as the secondary interface. The primary interface initiates RHPs. The ring takes effect in vlan 2.

Syntax: [no] metro-ring <*ring-id*>

The <*ring-id*> parameter specifies the ring ID 1 – 255. Configure the same ring ID on each of the nodes in the ring.

Syntax: [no] name <*string*>

The <*string*> parameter specifies a name for the ring. The name is optional, but it can be up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: "Customer A").

Syntax: [no] master

Configures this node as the master node for the ring. Enter this command only on one node in the ring. The node is a member (non-master) node by default.

Syntax: [no] ring-interface ethernet <*primary-if*> **ethernet** <*secondary-if*>

The **ethernet** <*primary-if*> parameter specifies the primary interface. On the master node, the primary interface originates RHPs. Ring control traffic will flow out of this interface by default. On member nodes the order in which you enter the interfaces does not matter as the secondary interface is determined by the receipt of RHP's from the master meaning the other interface defined in config becomes the primary. Once the ring is enabled the configuration entries on a member switch will reflect the ring direction no matter what order they are originally entered.

The **ethernet** <*secondary-if*> parameter specifies the secondary interface.

Syntax: [no] enable

The **enable** command enables the ring.

Changing the hello and preforwarding times

You can also change the RHP hello time and preforwarding time. To do so, enter commands such as the following.

```
NetIron(config-vlan-2-mrp-1)# hello-time 200
NetIron(config-vlan-2-mrp-1)# preforwarding-time 400
```

These commands change the hello time to 200 ms and change the preforwarding time to 400 ms.

Syntax: [no] hello-time <*ms*>

Syntax: [no] preforwarding-time <*ms*>

The <*ms*> specifies the number of milliseconds.

The hello time can be from 100 – 1000 (one second). The default hello time is 100 ms.

The preforwarding time can be from 200 – 5000 ms, and must be at least twice the value of the hello time and must be a multiple of the hello time. The default preforwarding time is 300 ms.

A change to the hello time or preforwarding time takes effect as soon as you enter the command.

NOTE
You can use MRP ring diagnostics to determine whether you need to change the hello time and preforwarding time. Refer to "Using MRP diagnostics".

Changing the scale timer

You are able to decrease MRP convergence time by changing the MRP scale timer tick from 100 ms to 50 ms. To do so, enter the following command:

```
NetIron(config)# scale-timer mrp
```

Syntax: [no] scale-timer mrp

Note: This command accepts no values and is put in place as it is shown above.

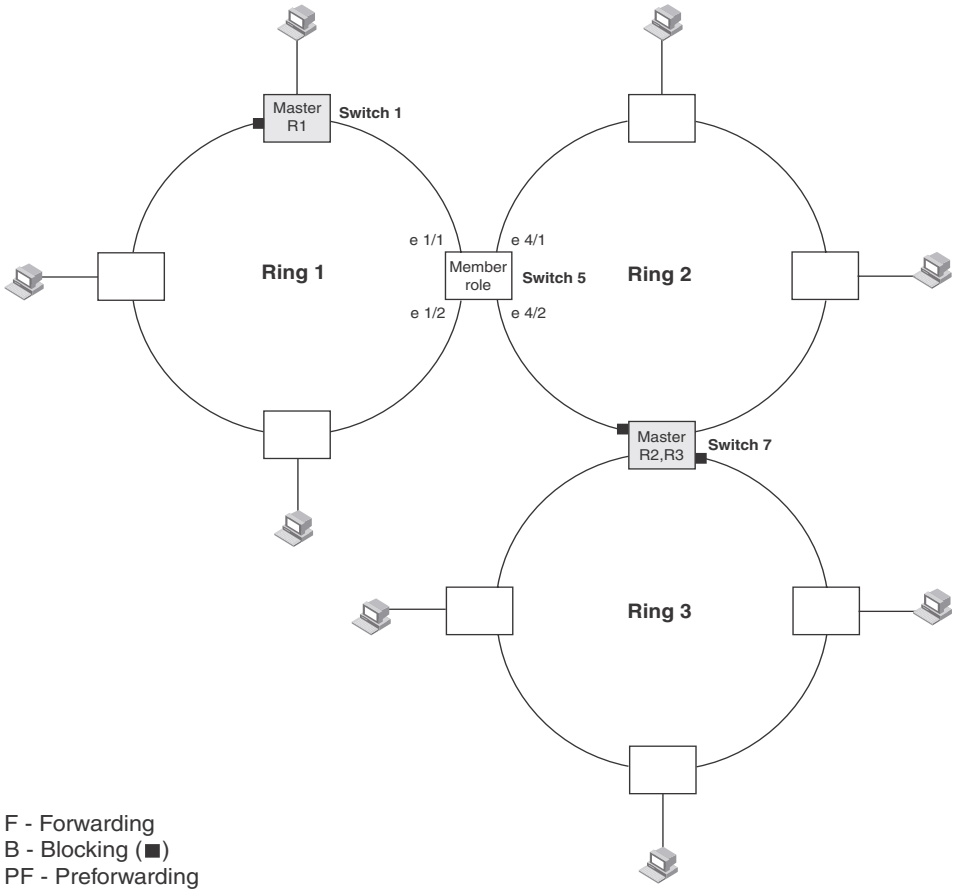
Note: Changing the scale timer affects the operation of MRP. Refer to "Tuning MRP timers" for further information.

MRP Phase 2

MRP phase 2 expands functionality by allowing a physical interface to be shared by multiple rings configured within the same vlan.

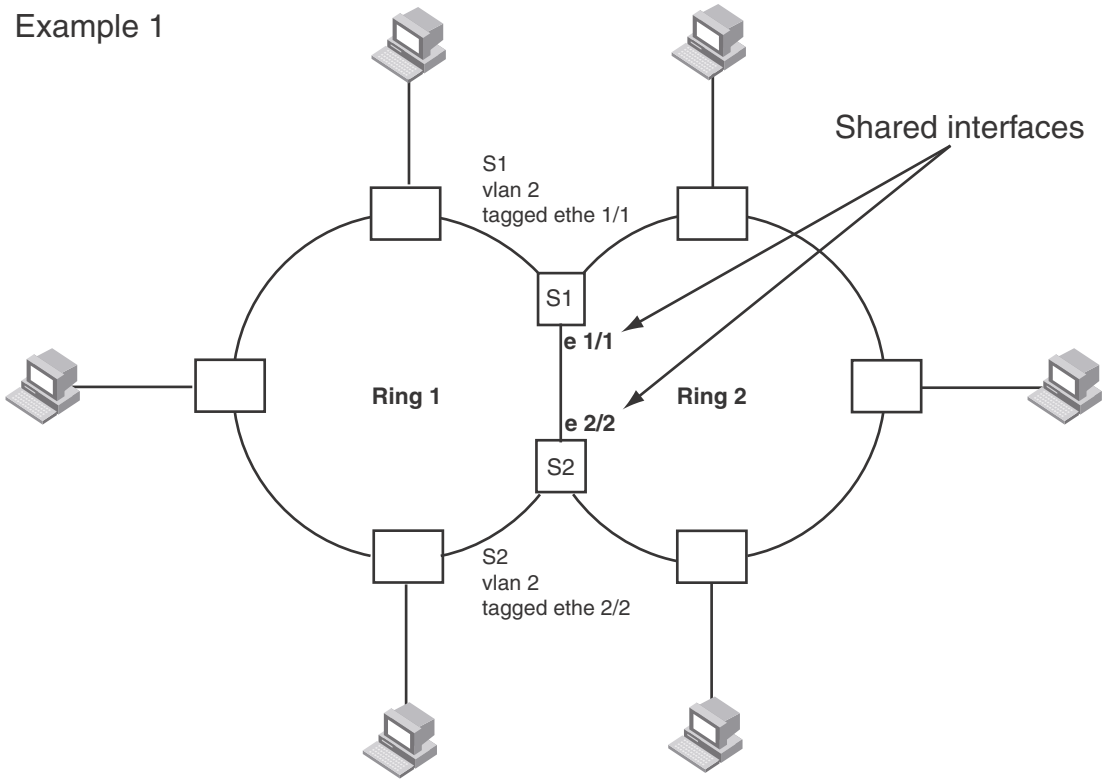
Recall that in MRP Phase 1, a node can have multiple MRP rings, but the rings cannot share the same interface. Any node can be designated as the master node for the ring. Each ring is an independent ring and RHP packets are processed within each ring exclusively.

FIGURE 61 Multiple MRP rings - phase 1



With MRP phase 2 multiple rings can be configured to share the same interface as long as the interfaces belong to the same vlan. Figure 62 shows an example of two rings that share the same interfaces on S1 and S2.

FIGURE 62 Example 1 multiple rings sharing interfaces - phase 2



On each node that will participate in ring 1, you configure the ring ID and the ring interfaces that will be used. You repeat the configuration steps for all nodes in ring 2. In a multiple ring configuration, a ring's ID determines its priority. The lower the ring ID, the higher the priority of a ring with ring ID 1 being the highest possible priority.

A key concept with MRP phase 2 is the ability to extend a single vlan across the whole topology even when multiple rings are required. Consider the example in Figure 63 where we have 3 MRP rings and a customer who needs to create neighbour relationships between all three routers depicted. The routers all have interfaces configured in a single subnet and need IP connectivity between each other.

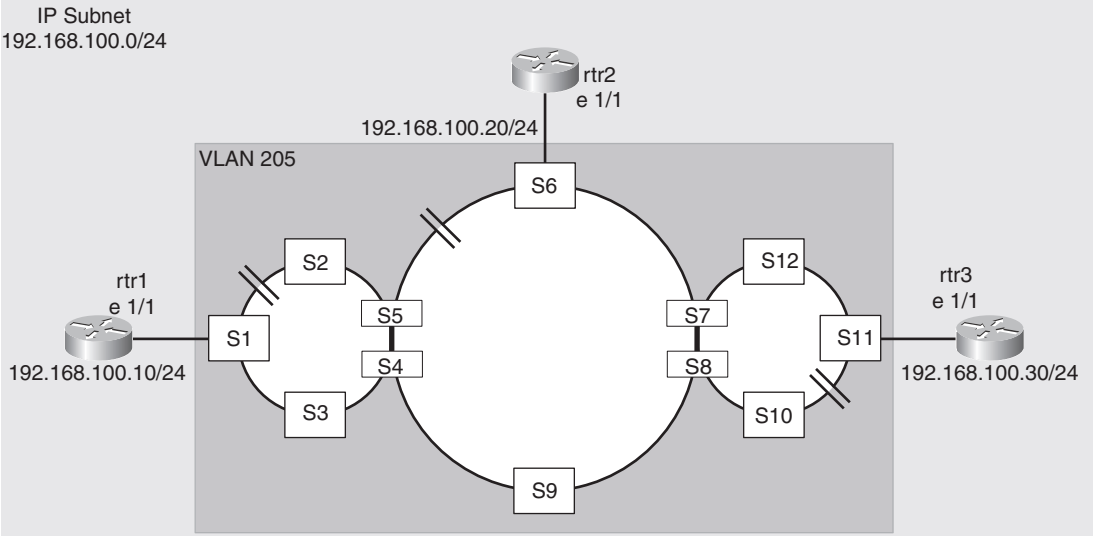
If each ring had an independant vlan then we would have to have a mechanism to move IP packets from a single IP subnet between different layer 2 topologies. By using MRP phase 2 we have multiple rings all associated with a single layer 2 topology allowing a common subnet to be distributed in the manner shown in the example. Whilst this looks nothing like a standard spanning tree network it should be treated in the same way from the perspective of a layer 2 topology, multiple paths where certain paths must be blocked to prevent loops at layer 2.

In addition it should be noted that the concept of multiple rings being associated with a single vlan describes the extent to which broadcasts will be propogated at layer 2 for that vlan. In other words a broadcast will be propogated to all ring interfaces in the layer 2 topology. The use of topology groups allows multiple vlans to effectively reuse a single layer 2 topology while maintaining a level of separation.

The obvious issue with this approach is that there must be a mechanism to prevent loops on the rings and this is the job of MRP, layer 2 loop prevention.

It is very easy to focus on the ring topology rather than the underlying layer 2 topology described by multiple rings. Design decisions are driven by the same factors as a standard spanning tree network replacing root bridges with ring masters. Traffic patterns at layer 2 are determined by which ring interfaces are forwarding and which are blocking and this in turn should drive design decisions for ring master placement as well as the direction of RHP flow from the ring masters. Traffic patterns in standard operation as well as failure mode can be determined prior to implementation allowing for appropriate capacity management on all links.

FIGURE 63 Multiple rings with one vlan spanning them



Ring interface ownership

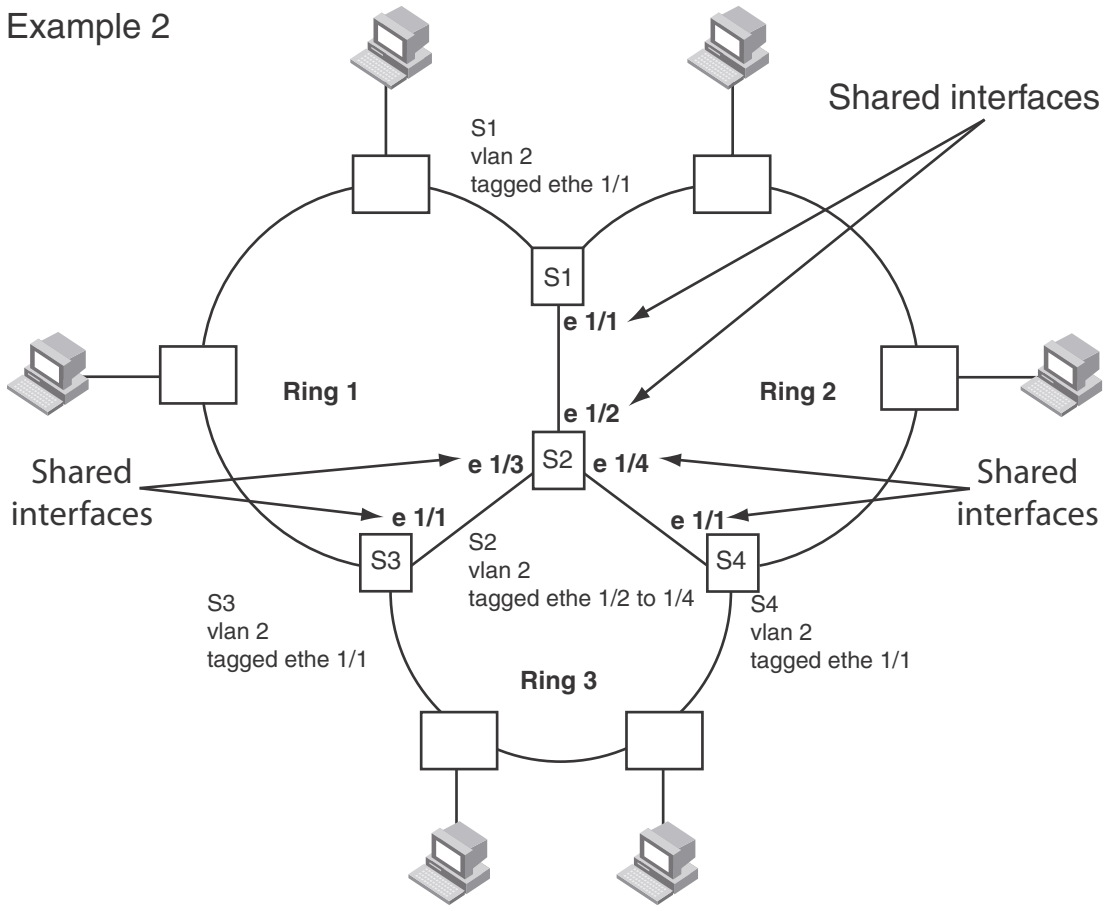
On a shared interface the highest priority ring will be the owner of the interface. In Figure 64 interface e 1/1 on S1 will be owned by ring 1 and marked as a regular interface while in ring 2 the same interface is marked as a tunnel interface in the output of the ‘show metro’ command.

On S2 interface e 1/2 is again owned by ring 1 and marked as a regular interface.

In Figure 64 the same principles of interface ownership apply. All shared interfaces on ring 1 nodes are shown as owned by ring 1 and marked as regular interfaces. Ring 2 will show shared interfaces as tunnel interfaces.

On S2 e 1/4 and S4 e 1/1 the interfaces will be owned by ring 2, as the highest priority ring on the interface, and ring 3 will show these interfaces as tunnel interfaces.

FIGURE 64 Example 2 multiple rings sharing interfaces - phase 2



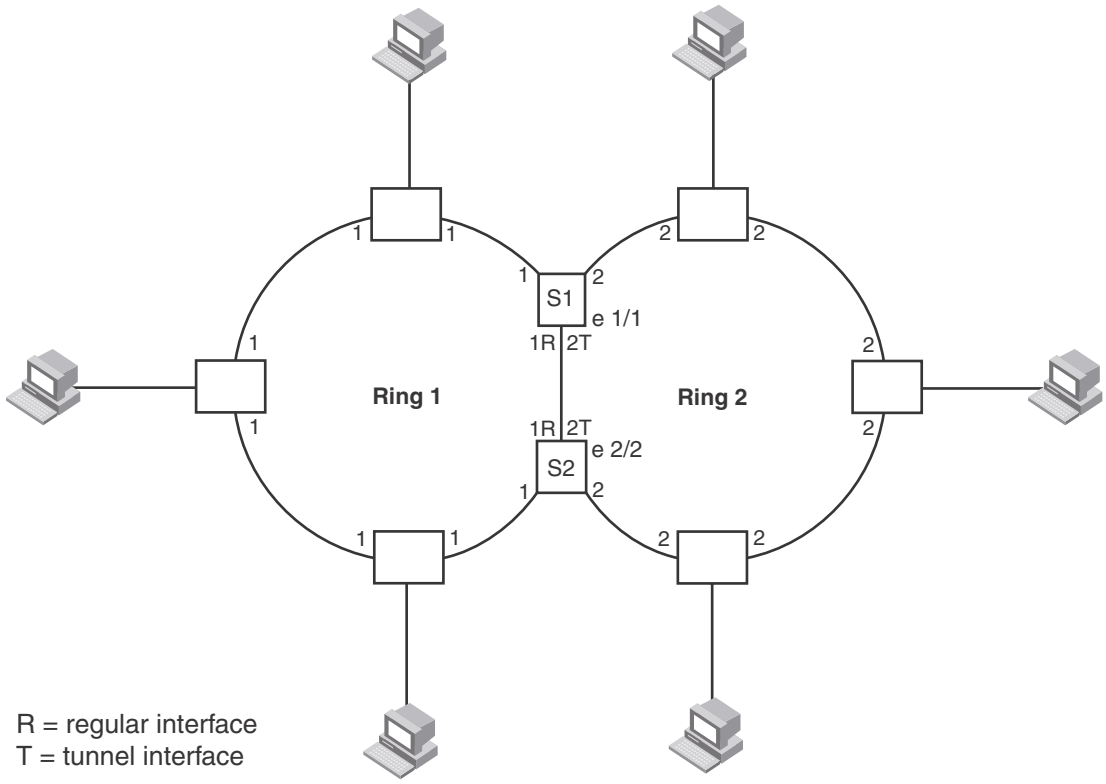
Ring interface IDs and types

For example, in Figure 65, all interfaces configured for ring 1 have a priority of 1. Interface e 1/1 on S1 and e 2/2 on S2 have a priority of 1 since 1 is the highest priority ring that shares the interface.

All interfaces on ring 2, except for e 1/1 on S1 and e 2/2 on S2 have a priority of 2.

If a node has shared interfaces then the ring interfaces that belong to the ring with the highest priority are regular interfaces for that ring and all lower priority ring interfaces are marked as tunnel interfaces. The highest priority ring configured becomes the priority for the interface.

FIGURE 65 Interface IDs and types



In Figure 65, nodes S1 and S2 have interfaces that belong to rings 1 and 2. Interface e 1/1 on S1 and e 2/2 on S2 are regular interfaces for ring 1, but they are tunnel interfaces for ring 2.

Selection of the master node for a ring

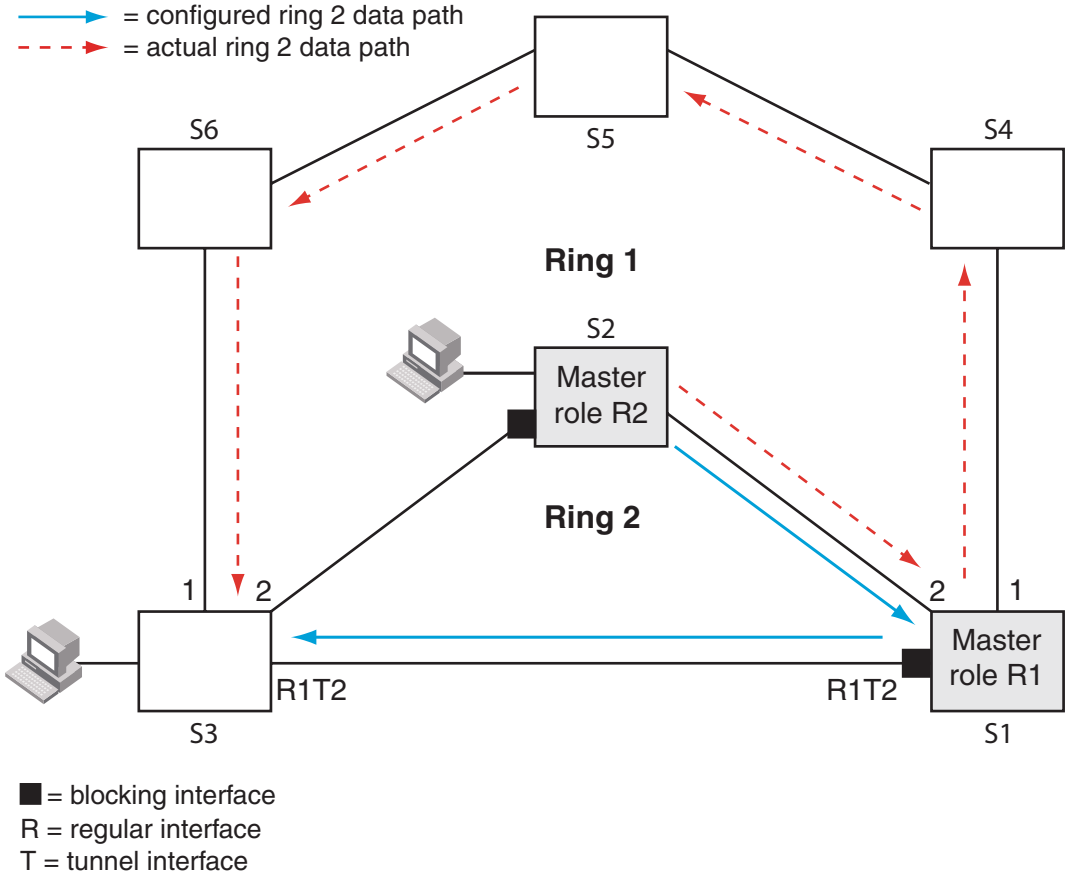
Configuring MRP rings with shared interfaces limits the nodes that can be designated as the master node for any particular ring.

- Any node on the ring that does not have any shared interfaces can be designated as the ring's master.
- You can only designate a node that has shared interfaces as master for a ring where all interfaces for the ring are marked as regular interfaces.
- On a node with shared interfaces, where you configure the role as master, the secondary ring interface should not be a shared interface. If you designate a shared interface as secondary it will be blocking under normal operation and allow RHP's but no data for lower priority rings. This can create unexpected traffic flows on the rings.

In Figure 65 any of the nodes on ring 1, even S1 or S2, can be a master node as all of the ring interfaces, even the shared interfaces between S1 and S2, are marked as regular interfaces for ring 1.

However for ring 2, neither S1 nor S2 can be a master node since the shared interfaces between S1 and S2 are marked as tunnel interfaces for ring 2.

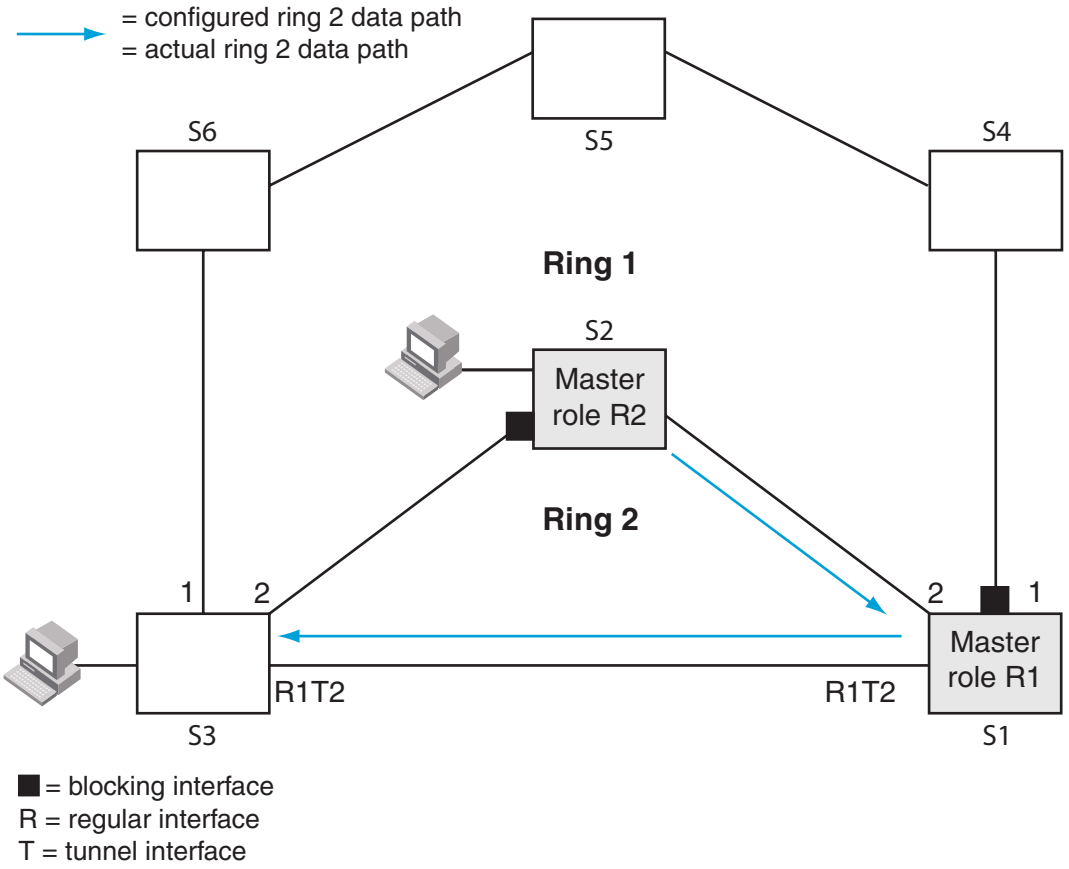
FIGURE 66 Unexpected switching path with shared interface



In Figure 66 ring 2 was configured with shared ring interfaces on S1 and S3 as depicted. S1 was configured as the master for ring 1 and the shared interface was defined as the secondary interface and subsequently blocks data. The designer intended the switching path between a host on S2 and another host on S3 to be via S1 shared interface, however due to the shared interface being blocked the actual switching path becomes S1 to S4,S5,S6 and finally S3.

Ring 2 is still operational but is not behaving in the manner which the design called for. By configuring the secondary interface on the regular port for ring 1 we obtain the expected result as shown in Figure 67.

FIGURE 67 Expected switching path with shared interface



RHP processing in rings with shared interfaces

Interfaces on an MRP ring have one of the following states:

- **Blocking (B)** – The interface can process RHPs, but cannot forward data for the ring. Only the secondary interface on the Master node can be blocking. If the interface receives RHP's for lower priority rings these RHPs will be discarded by this interface. This prevents RHP's from lower priority rings from looping in the topology.
- **Preforwarding (PF)** – The interface will forward RHPs but won't forward data for the ring. All ring interfaces start in this state when you enable MRP. A blocking interface transitions to preforwarding when the preforwarding timer expires.
- **Forwarding (F)** – The interface will forward RHP's and data for the ring. On member switches an interface transitions from preforwarding to forwarding when the preforwarding time expires or the interface receives an RHP with the forwarding bit set. A break in the ring is indicated if the secondary interface on the master fails to receive an RHP within the preforwarding timer and the interface transitions from blocking to forwarding to heal the ring. The preforwarding time is the number of milliseconds the interface will remain in the preforwarding state before changing to the Forwarding state, even without receiving an RHP.

The primary interface of the master node initiates RHP packets and sends them onto the ring. When the packet reaches a forwarding interface, MRP checks to see if the receiving interface is a regular interface or a tunnel interface:

- If the interface is a regular interface, the RHP packet is forwarded to the next interface. Forwarding of the packet continues on the ring until the secondary interface of the master node receives the packet and processes it. For the configured ring the receipt of an RHP with the same ring ID indicates the ring is healthy. RHPs for lower priority rings will be discarded without further processing at this point.
- If the interface is a tunnel interface, MRP checks the priority of the RHP packet and compares it to the priority of the tunnel interface:
  - If the RHP packet's priority is less than or equal to the interface's priority, the packet is forwarded through ring interfaces with higher priority which are in the forwarding state.
  - If the priority of the RHP packet is greater than the priority of the interface, the RHP packet is dropped. For example, if an RHP with a ring ID of 1 arrives at a tunnel interface owned by ring 2 the RHP will be dropped. If an RHP with a ring ID of 2 or 3 arrives at a tunnel interface owned by ring 2 the RHP will be forwarded.

NOTE
It is important to understand the key concept of RHPs leaking from lower priority rings to higher priority rings. Always remember that tunnel interfaces check the ring ID of an RHP before forwarding. Higher priority ring ID RHPs will be dropped.

How ring breaks are detected and healed between shared interfaces

If the link between shared interfaces breaks, the secondary interface on the highest priority ring master node changes to a preforwarding state, refer to Figure 70 on page 475. Any RHP from lower priority rings can traverse this interface and thus maintain the integrity of the lower priority rings. When the secondary interface changes state to forwarding the lower priority ring RHP's continue to traverse the interface.

This behaviour allows the ring 2 RHP's to continue around ring 1 and back to ring 2 until it reaches the secondary interface on ring 2's master node which changes to blocking mode since it receives its own RHP.

NOTE
On the ring member node, the primary and secondary interface is decided by the RHP flow from the ring master. The secondary interface is always the RHP receiver for its ring RHP's, the primary interface is always the sender of its rings RHP's. If there is no active ring master in the topology, then the running configuration on the member node will show exactly what was configured. This may change on introduction of an active ring master.

Normal flow

Figure 68 and Figure 69 show how RHP packets are forwarded in rings with shared interfaces. Figure 68 shows the flow of ring 1 RHPs while Figure 69 shows how ring 2 RHPs flow.

Interface e 2/1 is the primary interface of the ring 1 master node. The primary interface forwards an RHP packet on the ring. Since all the interfaces on ring 1 are regular interfaces, the RHP packet is forwarded until it reaches interface e 2/2, the secondary interface of the ring 1 master. Receipt of this RHP indicates a healthy ring 1 and interface e2/2 then changes to or maintains its state of blocking.

No copies of the ring 1 RHPs are forwarded on ring 2 tunnel interfaces or ring 2 regular interfaces in accordance with the rule that a higher priority RHP is not permitted to traverse a lower priority ring interface.

FIGURE 68 RHP flow on rings with shared interfaces showing ring 1 RHP flow

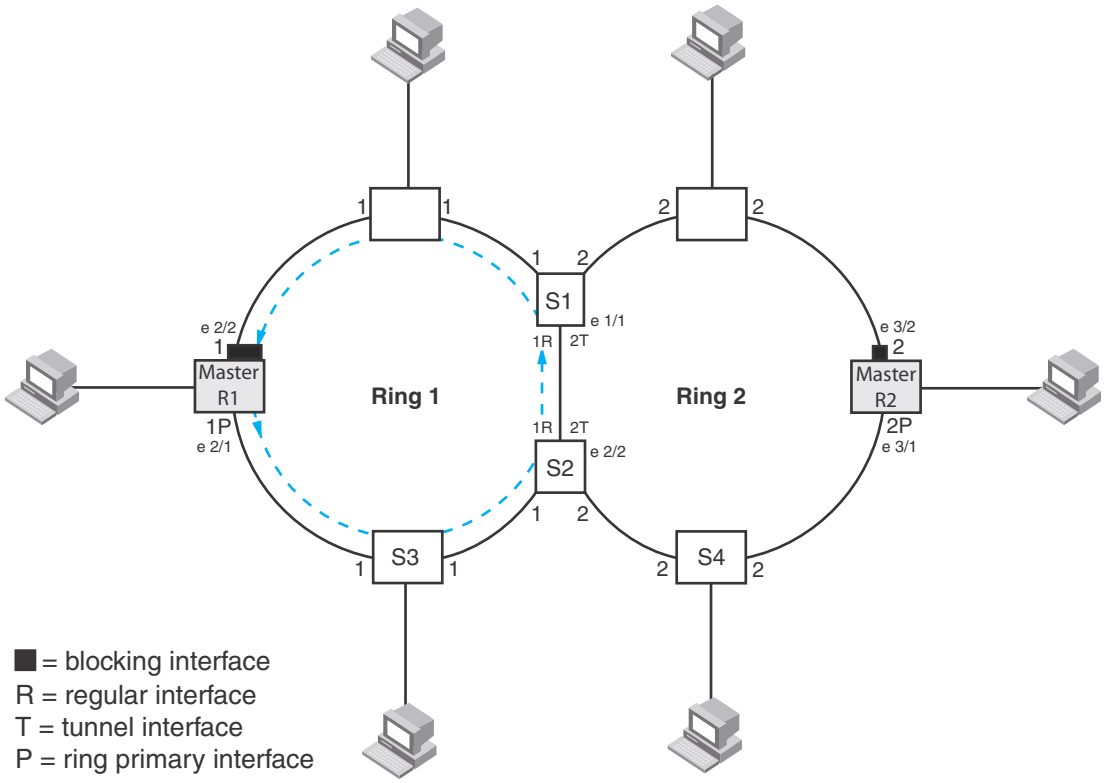
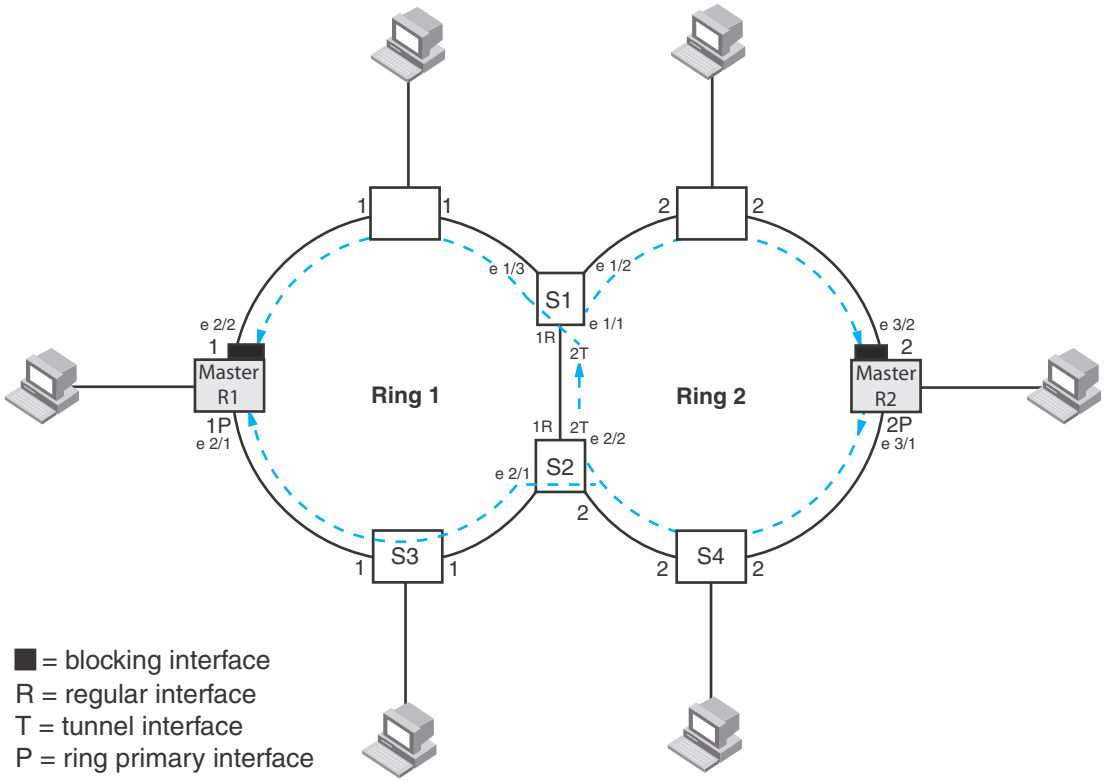


FIGURE 69 RHP flow on rings with shared interfaces showing ring 2 RHP flow



Referring to Figure 69 interface 3/1, is the primary interface of the ring 2 master node. It sends an RHP packet on the ring. Since all interfaces on S4 are regular interfaces, the RHP packet is forwarded on those interfaces.

When the RHP reaches S2:

- A copy of the RHP is sent out of regular interface e 2/1 onto ring 1. This is in accordance with the rule that a lower priority RHP can traverse a higher priority ring interface. This RHP is forwarded until it reaches the ring 1 master where it is discarded.
- A copy of the RHP is forwarded out of the ring 2 tunnel interface on e 2/2

The RHP is received by S1 on e 1/1 and then:

- A copy of the RHP is sent out of regular interface e 1/2 on ring 2
- A copy of the RHP is sent out of regulat interface e 1/3 on ring 1. This is in accordance with the rule that a lower priority RHP can traverse a higher priority ring interface. This RHP is forwarded until it reaches the ring 1 master where it is discarded.

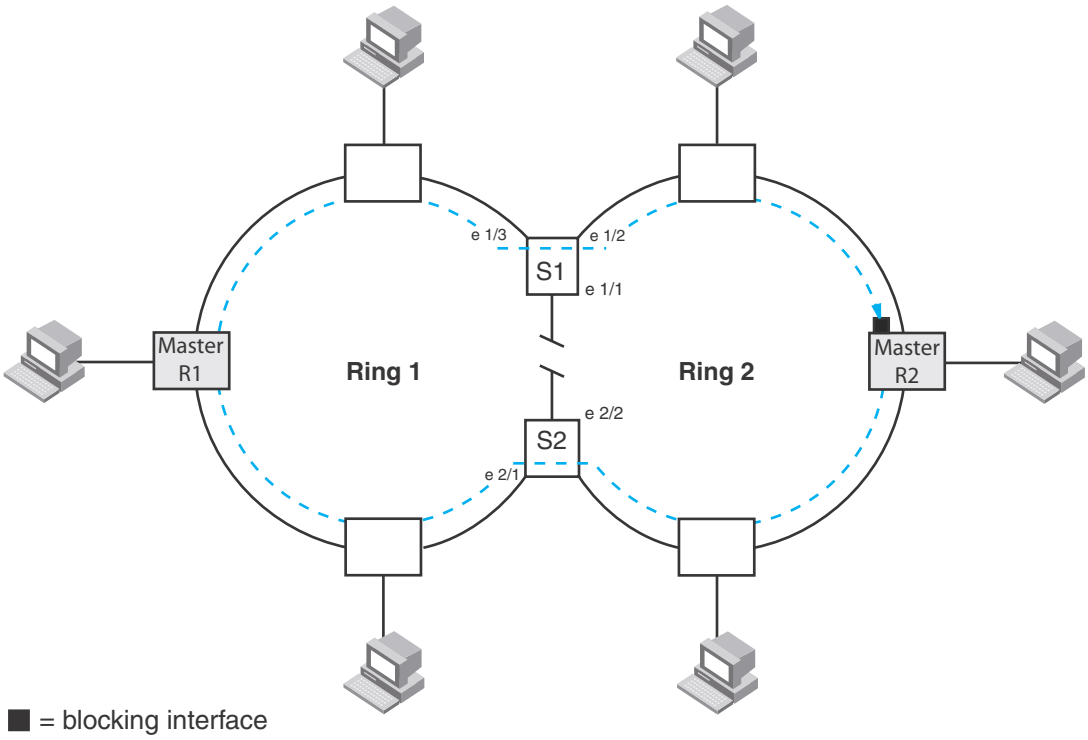
Flow when a link breaks

Referring to Figure 70 if the link between S1 and S2 fails, the secondary interface on the ring 1 master node changes to a forwarding state.

The RHPs from the master for ring 2 reach S2 and a copy of the RHP is forwarded out of e 2/1. This RHP traverses the ring 1 master and continues around ring 1 until it reaches S1. After S1 the RHP is back on ring 2 and is finally received by the master for ring 2 which keeps its secondary interface in blocking mode.

It should now be clear how the flow of lower priority RHPs over the higher priority ring ensure that both ring masters do not transition to forwarding and create a loop condition.

FIGURE 70 Flow of RHP packets when a link for shared interfaces breaks



Ring 2 RHPs follow this path until the link is restored. Once the link is restored the ring 1 master will transition its secondary ring interface to blocking and the ring 2 RHP flow is as shown in Figure 69.

NOTE
There should always be a layer 2 protocol configured in the default vlan when MRP is configured with all dual mode ports.

Configuring MRP with shared interfaces

MRP Phase 2 allows you to enter commands such as the following when configuring MRP.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# metro-ring 1
NetIron(config-vlan-2-mrp-1)# name CustomerA
NetIron(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
NetIron(config-vlan-2-mrp-1)# enable
```

```
NetIron(config-vlan-2-mrp-1)# metro-ring 2
NetIron(config-vlan-2-mrp-2)# name CustomerB
NetIron(config-vlan-2-mrp-2)# ring-interface ethernet 1/1 ethernet 1/2
NetIron(config-vlan-2-mrp-1)# enable
```

Syntax: [no] metro-ring <*ring-id*>

The <*ring-id*> parameter specifies the ring ID, which can be from 1 – 255. Configure the same ring ID on each of the nodes in the ring.

Syntax: [no] name <*string*>

The <*string*> parameter specifies a name for the ring. The name is optional, but it can have up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: “Customer A”).

Syntax: [no] ring-interface ethernet

The **ethernet** <*primary-if*> parameter specifies the primary interface. On the master node, the primary interface is the one that originates RHPs. Ring control traffic and layer 2 data traffic will flow in the outward direction from this interface by default. On member nodes, the direction of traffic flow depends on the traffic direction selected by the master node. Therefore, on a member node, the order in which you enter the interfaces does not matter.

The **ethernet** <*secondary-if*> parameter specifies the secondary interface.

Syntax: [no] enable

The **enable** command enables the ring.

Tuning MRP timers

To effectively tune MRP timers it is crucial to understand the association between the hello time and the preforwarding time.

Flushing the mac table following an MRP event

After an MRP event switches in the ring flush MAC tables and relearn to ensure correct forwarding paths. Notification to flush is carried out by sending topology change RHP’s.

Hello time

This timer specifies the interval at which RHP’s are generated by the ring master. It should be noted that this interval is applied not only to standard RHP’s but also to topology change notification RHP’s. For example: Setting the hello time to its maximum value of 15,000 ms would mean that the three topology change notification RHP’s that are sent following a ring break being detected or a ring heal event would result in Mac table flushes three times at 15 second intervals. On a busy network this would cause unnecessary impact.

Preforwarding time

The preforwarding time defines the amount of time an interface will take to move from blocking to preforwarding without RHP’s being received. It also defines the amount of time an interface will take to move from preforwarding to forwarding without RHP’s being received.

The preforwarding time must be at least 2 x hello time and must be a multiple of the hello time.

The preforwarding time for a lower priority ring must be greater than or equal to the highest higher priority ring.

For example: Setting the preforwarding time to its maximum value of 30,000 ms will mean that a break in the ring (assuming no alarm RHP's are generated) will take one minute to heal.

Setting hello and preforwarding timers appropriately

When setting timers both the hello time and the preforwarding time should be considered to ensure that the appropriate recovery time is applied on the network.

Consider a break in the network that does not generate alarm RHP's

Example 1(default values):

Preforwarding time = 300ms

Hello time = 100ms

Time to forwarding = 2 x preforwarding time = 600ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 900ms = 0.9secs

Example 2:

Preforwarding time = 10000ms

Hello time = 100ms

Time to forwarding = 2 x preforwarding time = 20000ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 20300ms = 20.3secs

Example 3:

Preforwarding time = 10000ms

Hello time = 5000ms

Time to forwarding = 2 x preforwarding time = 20000ms

Post recovery mac table flush time = 3 x hello time = 15000ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 35000ms = 35secs

It can therefore be seen that the hello time should not be changed on the network unless there is evidence of regular misses on the ring. Time to traverse the ring can be determined by running MRP diagnostics.

Effect of the scale timer

Changing the scale timer has a significant effect on the operation of MRP and should be considered for very high performance low latency networks where a very rapid failure detection and recovery mode is required. Achieving this rapid detection and recovery requires very stable high speed environments to prevent a high level of unnecessary topology changes in the environment.

The effect of setting the scale timer is that the time taken to move from blocking to preforwarding and preforwarding to forwarding is (preforwarding value – the hello time). This is a significant change to the operation of MRP in the default state which has been described in the previous section.

Note: When setting the timer at the CLI the actual value used will be exactly half of the input value. The examples that follow assume the corrected value.

Consider a break in the network that does not generate alarm RHP's

Example 1(default values):

Preforwarding time = 300ms

Hello time = 100ms

Time to forwarding = preforwarding time – hello time = 200ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 500ms = 0.5secs

Example 2:

Preforwarding time = 100ms

Hello time = 50ms

Time to forwarding = preforwarding time – hello time = 50ms

Post recovery mac table flush time = 3 x hello time = 150ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 200ms = 0.2secs

Example 3:

Preforwarding time = 10000ms

Hello time = 5000ms

Time to forwarding = preforwarding time – hello time = 5000ms

Post recovery mac table flush time = 3 x hello time = 15000ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 20000ms = 20secs

Using MRP diagnostics

The MRP diagnostics feature calculates how long it takes for RHP packets to travel through the ring. When you enable MRP diagnostics, the software tracks RHP packets according to their sequence numbers and calculates how long it takes an RHP packet to travel one time through the entire ring. When you display the diagnostics, the CLI shows the average round-trip time for the RHP packets sent since you enabled diagnostics. The calculated results have a granularity of 1 microsecond.

Enabling MRP diagnostics

To enable MRP diagnostics for a ring, enter the following command on the Master node, at the configuration level for the ring.

```
NetIron(config-vlan-2-mrp-1)#diagnostics
```

Syntax: [no] diagnostics

NOTE
When using the 'show metro' command, the member node of a ring does not display correctly since the MRP RHPs are hardware forwarded (or software forwarded on the linecard), these statistics are only reflective of the MRP RHPs that made it to the management processor. In most cases, these would be TC RHPs since the MP needs to flush MACs in that case.

NOTE
This command is valid only on the master node.

Displaying MRP diagnostics

To display MRP diagnostics results, enter the following command on the Master node.

```
NetIron(config)# show metro 2 diag

Metro Ring 2 - CustomerA
=============
diagnostics results

Ring       Diag       RHP average     Recommended     Recommended
id         state      time(microsec)  hello time(ms)  Prefwing time(ms)
2          enabled    125             100             300

Diag frame sent     Diag frame lost
1230                0
```

Syntax: show metro <*ring-id*> **diag**

This display shows the following information.

TABLE 90 CLI display of MRP ring diagnostic information

This field...	Displays...
Ring id	The ring ID.
Diag state	The state of ring diagnostics.

TABLE 90 CLI display of MRP ring diagnostic information (Continued)

This field...	Displays...
RHP average time	The average round-trip time for an RHP packet on the ring. The calculated time has a granularity of 1 microsecond.
Recommended hello time	The hello time recommended by the software based on the RHP average round-trip time.
Recommended Prefwing time	The preforwarding time recommended by the software based on the RHP average round-trip time.
Diag frame sent	The number of diagnostic RHPs sent for the test.
Diag frame lost	The number of diagnostic RHPs lost during the test.

If the recommended hello time and preforwarding time are different from the actual settings and you want to change them, refer to "Configuring MRP" on page 462.

Displaying MRP information

You can display the following MRP information:

- Topology group ID associated with the MRP ring
- Ring configuration information and statistics

Displaying topology group information

To display topology group information, enter the following command.

Syntax: show topology-group [<*group-id*>]

Refer to "Displaying topology group information" on page 547 for more information.

Displaying ring information

To display ring information, enter the following command.

```
NetIron(config)# show metro
Metro Ring 10 - VLAN Type REGULAR
==============
Ring       State      Ring       Master     Topo       Hello      Prefwing
id                    role       vlan       group      time(ms)   time(ms)
10         enabled    member     7          1          100        300

Ring interfaces Interface role  Interface state interface type
ethernet 1/1    primary         forwarding      regular
ethernet 30/1   secondary       forwarding      regular

RHPs sent       RHPs rcvd       TC rcvd    TC sent    State changes
0               0               69         0          6
```

Syntax: show metro [<*ring-id*>]

This display shows the following information.

TABLE 91 CLI display of MRP ring information

This field...	Displays...
Ring id	The ring ID
State	The state of MRP. The state can be one of the following: • enabled – MRP is enabled • disabled – MRP is disabled
Ring role	Whether this node is the master for the ring. The role can be one of the following: • master • member
Topo group	The topology group ID if a topology group is configured. This field will show the value 'not conf' if no topology group is in use.
Hello time	The interval, in milliseconds, which the forwarding port on the ring's master node sends Ring Hello Packets (RHPs). Configured in increments of 100ms.
Prefwing time	The number of milliseconds a MRP interface will wait to move an interface in blocking state to preforwarding state if no RHP's are received. It is also the number of milliseconds that an interface that has entered the preforwarding state will wait before changing to the forwarding state. If a member port in the preforwarding state does not receive an RHP within the preforwarding time (Prefwing time), the port assumes that a topology change has occurred and changes to the forwarding state. The secondary port on the master node changes to blocking if it receives an RHP, but changes to forwarding if the port does not receive an RHP before the preforwarding time expires. Configured in increments of 100ms. **NOTE:** A member node's preforwarding interface also changes from preforwarding to forwarding if it receives an RHP whose forwarding bit is on.
Ring interfaces	The device's two interfaces with the ring. **NOTE:** If the interfaces are trunk groups, only the primary ports of the groups are listed.

Interface role	The interface role can be one of the following:
	• primary
	  • Master node – The interface generates RHPs.
	  • Member node – The interface forwards RHPs received on the other interface (the secondary interface).
	• secondary – The interface does not generate RHPs.
	  • Master node – The interface listens for RHPs.
	  • Member node – The interface receives RHPs.

Forwarding state	Whether MRP Forwarding is enabled on the interface. The forwarding state can be one of the following: • blocking – The interface is blocking layer 2 data traffic and RHPs • disabled – The interface is down • forwarding – The interface is forwarding layer 2 data traffic and RHPs • preforwarding – The interface is listening for RHPs but is blocking layer 2 data traffic
Interface Type	Shows if the interface is a regular port or a tunnel port.
RHPs sent	The number of RHPs sent on the interface.
RHPs rcvd	The number of RHPs received on the interface.

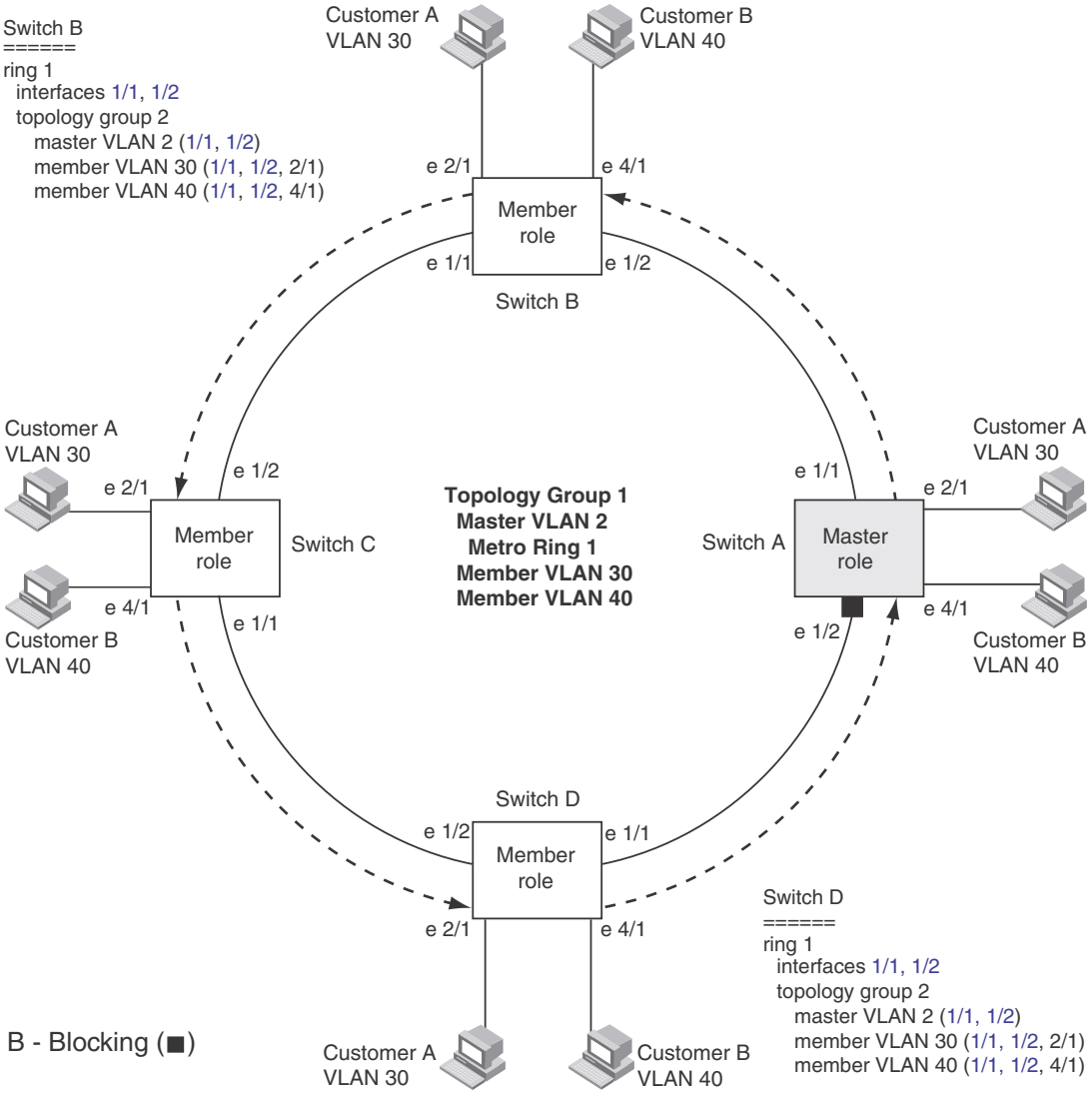
TABLE 91 CLI display of MRP ring information (Continued)

This field...	Displays...
TC RHPs rcvd	The number of Topology Change RHPs received on the interface.
State changes	The number of MRP interface state changes that have occurred. The state can be one of the states listed in the Forwarding state field.

MRP CLI example

The following examples show the CLI commands required to implement the MRP configuration shown in Figure 71.

FIGURE 71



NOTE
For simplicity, the figure shows the vlans on only two switches. The CLI examples implement the ring on all four switches.

Commands on Switch A (master node)

The following commands configure a vlan for the ring. The ring vlan must contain both of the node's interfaces with the ring. Add these interfaces as tagged interfaces, since the interfaces also must be in each of the customer vlans configured on the node.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-2)# metro-ring 1
NetIron(config-vlan-2-mrp-1)# name "Metro A"
NetIron(config-vlan-2-mrp-1)# master
NetIron(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
NetIron(config-vlan-2-mrp-1)# enable
NetIron(config-vlan-2-mrp-1)# exit
NetIron(config-vlan-2)# exit
```

The following commands configure the customer vlans. The customer vlans must contain both the ring interfaces as well as the customer interfaces.

```
NetIron(config)# vlan 30
NetIron(config-vlan-30)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-30)# tag ethernet 2/1
NetIron(config-vlan-30)# exit
NetIron(config)# vlan 40
NetIron(config-vlan-40)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-40)# tag ethernet 4/1
NetIron(config-vlan-40)# exit
```

The following commands configure topology group 1 on vlan 2. The master vlan is the one that contains the MRP configuration. The member vlans use the MRP parameters of the master vlan. The control interfaces (the ones shared by the master vlan and member vlan) also share MRP state.

```
NetIron(config)# topology-group 1
NetIron(config-topo-group-1)# master-vlan 2
NetIron(config-topo-group-1)# member-vlan 30
NetIron(config-topo-group-1)# member-vlan 40
```

Commands on Switch B

The commands for configuring switches B, C, and D are similar to the commands for configuring switch A, with two differences: the nodes are not configured to be the ring master. Omitting the **master** command is required for non-master nodes.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-2)# metro-ring 1
NetIron(config-vlan-2-mrp-1)# name "Metro A"
NetIron(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
NetIron(config-vlan-2-mrp-1)# enable
NetIron(config-vlan-2)# exit
```

```
NetIron(config)# vlan 30
NetIron(config-vlan-30)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-30)# tag ethernet 2/1
NetIron(config-vlan-30)# exit
NetIron(config)# vlan 40
NetIron(config-vlan-40)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-40)# tag ethernet 4/1
NetIron(config-vlan-40)# exit
NetIron(config)# topology-group 1
NetIron(config-topo-group-1)# master-vlan 2
NetIron(config-topo-group-1)# member-vlan 30
NetIron(config-topo-group-1)# member-vlan 40
```

Commands on Switch C

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-2)# metro-ring 1
NetIron(config-vlan-2-mrp-1)# name “Metro A”
NetIron(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
NetIron(config-vlan-2-mrp-1)# enable
NetIron(config-vlan-2)# exit
NetIron(config)# vlan 30
NetIron(config-vlan-30)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-30)# tag ethernet 2/1
NetIron(config-vlan-30)# exit
NetIron(config)# vlan 40
NetIron(config-vlan-40)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-40)# tag ethernet 4/1
NetIron(config-vlan-40)# exit
NetIron(config)# topology-group 1
NetIron(config-topo-group-1)# master-vlan 2
NetIron(config-topo-group-1)# member-vlan 30
NetIron(config-topo-group-1)# member-vlan 40
```

Commands on Switch D

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-2)# metro-ring 1
NetIron(config-vlan-2-mrp-1)# name “Metro A”
NetIron(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
NetIron(config-vlan-2-mrp-1)# enable
NetIron(config-vlan-2)# exit
NetIron(config)# vlan 30
NetIron(config-vlan-30)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-30)# tag ethernet 2/1
NetIron(config-vlan-30)# exit
NetIron(config)# vlan 40
NetIron(config-vlan-40)# tag ethernet 1/1 to 1/2
NetIron(config-vlan-40)# tag ethernet 4/1
NetIron(config-vlan-40)# exit
NetIron(config)# topology-group 1
NetIron(config-topo-group-1)# master-vlan 2
NetIron(config-topo-group-1)# member-vlan 30
NetIron(config-topo-group-1)# member-vlan 40
```

Configuring MRP under an ESI VLAN

MRP can also be configured under a vlan that is part of a user-configured ESI. Configuring MRP in this scenario is exactly the same as explained before.

```
NetIron(config)# esi customer1 encapsulation cvlan
NetIron(config-esi-customer1)# vlan 100
NetIron(config-esi-customer1-vlan-100)# tag ethernet 1/1 to 1/2
NetIron(config-esi-customer1-vlan-100)# metro-ring 1
NetIron(config-esi-customer1-vlan-100-mrp-1)# name "Metro A"
NetIron(config-esi-customer1-vlan-100-mrp-1)# master
NetIron(config-esi-customer1-vlan-100-mrp-1)# ring-interface ethernet 1/1
ethernet 1/2
NetIron(config-esi-customer1-vlan-100-mrp-1)# enable
NetIron(config-esi-customer1-vlan-100-mrp-1)# exit
NetIron(config-esi-customer1-vlan-100)# exit
```

Configuration considerations

The configuration considerations are as follows:

- MRP can be configured for vlans with encapsulation type B-VLAN, S-VLAN or C-VLAN.
- When MRP is configured for vlans under an ESI, the MRP members must be part of the same ESI.

Ethernet Ring Protection Protocol 14

The following Ethernet Ring Protection (ERP) features are supported by .

- Signal fail
- Manual switch
- Forced switch
- Dual-end blocking
- Non-revertive mode
- Interconnected ring
- FDB flush optimization

Ethernet Ring Protection

Ethernet Ring Protection (ERP), a non-proprietary protocol described in ITU-T G.8032 (Version 1 and 2), integrates an Automatic Protection Switching (APS) protocol and protection switching mechanisms to provide Layer 2 loop avoidance and fast reconvergence in Layer 2 ring topologies. ERP supports multi-ring and ladder topologies. ERP can also function with IEEE 802.1ag to support link monitoring when non-participating devices exist within the Ethernet ring.

You can enable one instance of ERP on a device. Changes to a master VLAN apply to the member VLANs.

NOTE
Before configuring ERP, you must configure a VLAN and the ports you require for your deployment.

This chapter describes ERP components, features, and how to configure, and manage ERP.

Ethernet Ring Protection components

An ERP deployment consists of the following components:

- Roles assigned to devices, called Ethernet Ring Nodes (ERN)
- Interfaces
- Protocols — ERP alone or with IEEE 802.1ag
- ERP messaging
- ERP operational states
- ERP timers

ERN roles

In an Ethernet ring topology you can assign each ERN one of three roles:

- **Ring Protection Link Owner (RPL owner)** — One RPL owner must exist in each ring; its role is to prevent loops by maintaining a break in traffic flow to one configured link while no failure condition exists within the ring.
- **Non-RPL node** — Multiple non-RPL nodes, can exist in a ring; but they have no special role and perform only as ring members. Ring members apply and then forward the information received in R-APS messages.
- **Ring Protection Link (RPL) node** — RPL nodes block traffic to the segment that connects to the blocking port of the RPL owner. The RPL node is used in dual-end blocking and is part of the FDB optimization feature.

Each device can only have one role at any time. Non-ERN devices can also exist in topologies that use IEEE 802.1ag.

ERN interfaces

In addition to a role, each ERN has two configured interfaces:

- Left interface
- Right interface

Traffic enters one interface (ingress) and exits the device using the other interface (egress). The right and left interfaces are physically connected.

You must configure these left and right interfaces in the same pattern across all ERNs within a topology. For example you can assign the interfaces as left/right, left/right, left/right, and so on. It is not acceptable, however, to assign interfaces in random order, such as left/right in the configuration of one ERN and then right/left in the configuration of the next ERN.

Protocols

You can configure standalone ERP or ERP with IEEE 802.1ag support.

Using standalone ERP

When using standalone ERP, all devices have a role, and all devices participate at least as ERP members.

Ring-APS (R-APS) messages are sent at initial start-up of a configuration and periodically when link or node failures or recoveries occur. Each ERN applies the information received in the R-APS messages and forwards the received RAPS messages if both ports are in the forwarding state.

The sending ERN terminates the message when it receives a message originally sent from itself.

Configurable timers prevent ERNs from receiving outdated messages and decrease failure reporting time to allow increased stability within the topology.

To properly configure and troubleshoot ERP, an understanding of the messaging, operational states, and timers is essential. For more information about the ERP protocol, see ITU-T G.8032.

Using ERP with IEEE 802.1ag support

When you have other nonparticipating switches in the ring, you can use the IEEE 802.1ag support to perform link health checks to the next ERN.

With IEEE 802.1ag configured, the ERNs within the ring send Continuity Check Messages (CCM) to verify the integrity of their own links. If a node is not receiving CCMs or if a link goes down, a failure is reported to the ring through R-APS messages. See Figure 72.

FIGURE 72 ERP with IEEE 802.1ag support

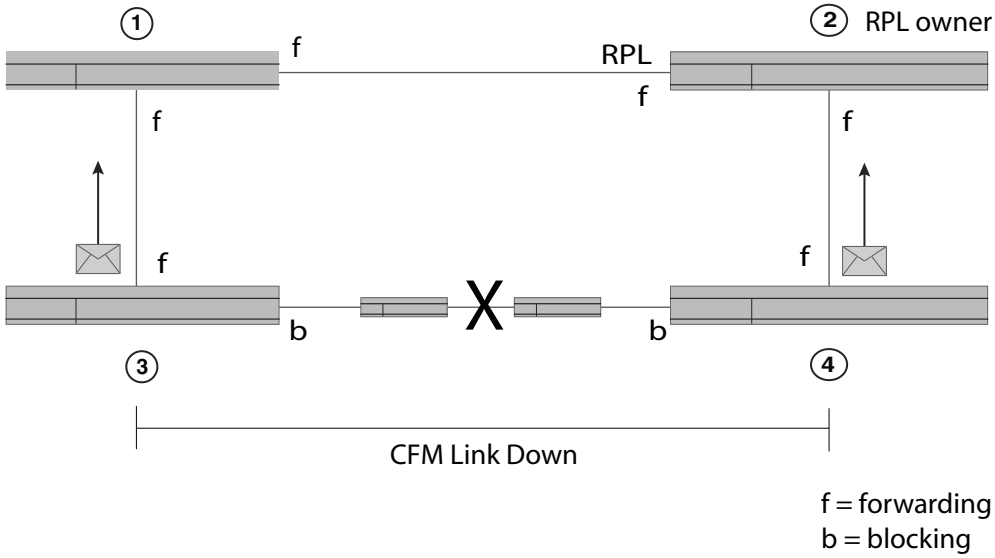


Figure 72 shows a segment with ERNs 3 and 4 and two non-participating switches located on the same network segment between them. When ERNs 3 and 4 stopped receiving CCMs, the following actions occurred on ERNs 3 and 4:

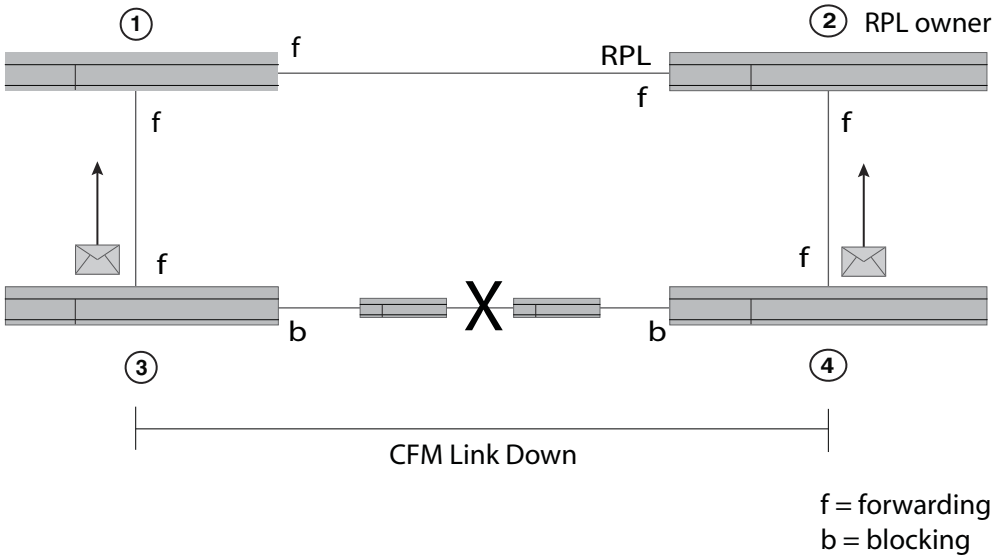
1. Blocked the failed port
2. Transmitted a R-APS (SF) message
3. Unblocked the non-failed port
4. Flushed the FDB
5. Entered the Protection state

As a result, ERN 2, the RPL owner, unblocked the RPL, and the topology became stable and loop free.

ERP messaging

In ERP, ERNs send R-APS messages. Figure 73 shows the general R-APS packet structure. For details about the packet structures, see ITU-T G.8032.

FIGURE 73 R-APS packet structure



The destination MAC address (Dst Mac) is the first element in the packet and is of the form 01-19-A7-00-00-<ERP ID>. The default value is 01. However, you can configure the ERP ID with the **raps-default-mac** command. In ITU-T G.8032 Version 1 the default value is always used.

The Node ID indicates the base MAC address and can be found in the R-APS specific information part of a R-APS message.

ERP operational states

RPL nodes can be in one of six different states in Version 2:

- Init
- Idle
- Protection state, which is designated as a signal fail (SF) event in the R-APS
- Manual-switch (MS)
- Forced-switch (FS)
- Pending (not available if using Version 1)

When an ERP topology starts up, each ERN (in Init state) transmits a R-APS (NR). After start-up, the behavior varies by assigned role. Figure 74 on page 491 shows the initialization process for an ERN.

FIGURE 74 Message exchange and actions during ERN initialization version 2

RPL owner	Non-RPL node	RPL node
Init state	Init state	Init state
1 Blocks the RPL 2 Sends a R-APS (NR) 3 Enters the Pending state.	1 Blocks the left interface 2 Sends a R-APS (NR) 3 Enters the Pending state	1 Blocks the left interface 2 Sends a R-APS (NR) 3 Enters the Pending state
4. Starts the WTR timer 5. (After the WTR expires) stops sending NR 6. Sends R-APS (NR, RB, DNF) 7. Enters the Idle state	After receiving the (NR, RB, DNF) from the RPL owner: 1 Unblocks the non-failed blocking port 2 Stops sending (NR) 3 Enters the Idle state	After receiving the (NR, RB, DNF) from the RPL owner: 1 Blocks the RPL port 2 Unblocks the other ports 3 Enters the Idle state

When the ring is in the Pending state, an ERN flushes the filtering database (FDB) if it receives any of the following state requests:

- Signal-fail (SF)
- No request (NR), RPL Blocked (RB)

NOTE
ITU-T G.8032 Version 1 does not use a Pending state, so from the Protection state ERNs enter the Idle state.

ERP timers

ERP provides various timers to ensure stability in the ring while a recovery is in progress or to prevent frequent triggering of the protection switching. All of the timers are operator configurable.

- **Guard timer** — All ERNs use a guard timer. The guard timer prevents the possibility of forming a closed loop and prevents ERNs from applying outdated R-APS messages. The guard timer activates when an ERN receives information about a local switching request, such as after a switch fail (SF), manual switch (MS), or forced switch (FS). When this timer expires, the ERN begins to apply actions from the R-APS it receives. This timer cannot be manually stopped.
- **Wait to restore (WTR) timer** — The RPL owner uses the WTR timer. The WTR timer applies to the revertive mode to prevent frequent triggering of the protection switching due to port flapping or intermittent signal failure defects. When this timer expires, the RPL owner sends a R-APS (NR, RB) through the ring.
- **Wait to Block (WTB) timers** — This wait-to-block timer is activated on the RPL owner. The RPL owner uses WTB timers before initiating an RPL block and then reverting to the idle state after operator-initiated commands, such as for FS or MS conitions, are entered. Because multiple FS commands are allowed to co-exist in a ring, the WTB timer ensures that the clearing of a single FS command does not trigger the re-blocking of the RPL. The WTB timer is defined to be 5 seconds longer than the guard timer, which is enough time to allow a reporting ERN to transmit

two R-APS messages and allow the ring to identify the latent condition.

When clearing a MS command, the WTB timer prevents the formation of a closed loop due to the RPL owner node applying an outdated remote MS request during the recovery process.

- **Hold-off timer** — Each ERN uses a hold-off timer to delay reporting a port failure. When the timer expires, the ERN checks the port status. If the issue still exists, the failure is reported. If the issue does not exist, nothing is reported.
- **Message interval** — This is an operator configurable feature for sending out R-APS messages continuously when events happen.

Initializing a new ERN

A newly configured Version 2 ERP topology with four ERNs initializes as described in this section. The ERNs have the following roles:

- ERN 2 is the RPL owner.
- ERNs 1, 3, and 4 are non-RPL nodes.

Figure 75 shows the first step of initialization beginning from ERN 4, a non-RPL node. The actions of each ERN are:

- ERN 1 takes no action. Both ports are in the forwarding state.
- ERN 2 (RPL owner) takes no action. Both ports, including the RPL port, are in the VLAN port forwarding state.
- ERN 3 takes no action. Both ports are in the forwarding state.
- From the Init state ERN 4 stops all timers (guard, WTR, WTB), blocks the left port, unblocks the right port, transmits R-APS (NR) messages, and enters the Pending state.

FIGURE 75 Initializing an ERN topology - I

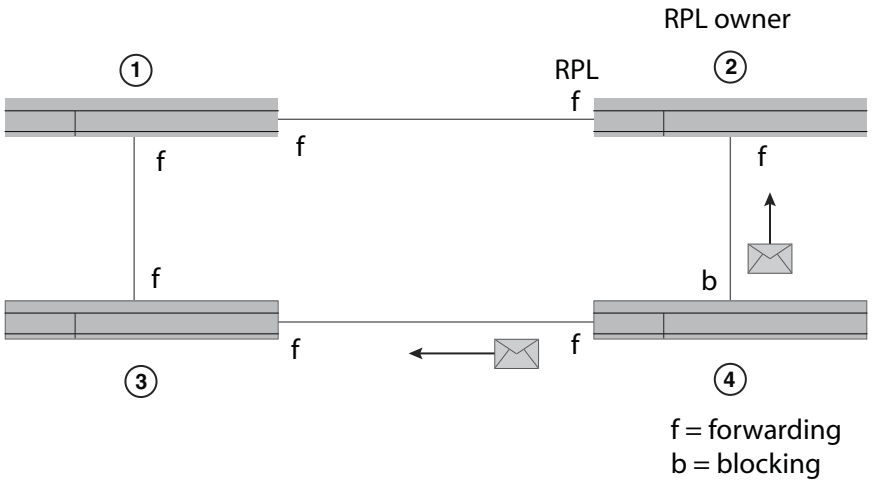


Figure 76 on page 493 shows the next sequence of events. Next, ERN 1 initializes. The actions of each ERN are:

- ERN 1 stops all timers (guard, WTR, WTB), blocks the left port, unblocks the right port, transmits R-APS (NR) messages, and enters the Pending state.

- ERN 2 takes no action. Both ports are in the forwarding state.
- ERN 3 takes no action. Both ports are in the forwarding state.
- ERN 4 stays in the Pending state, transmits R-APS (NR) messages, and continues to block the left interface.

FIGURE 76 Initializing an ERP topology - II

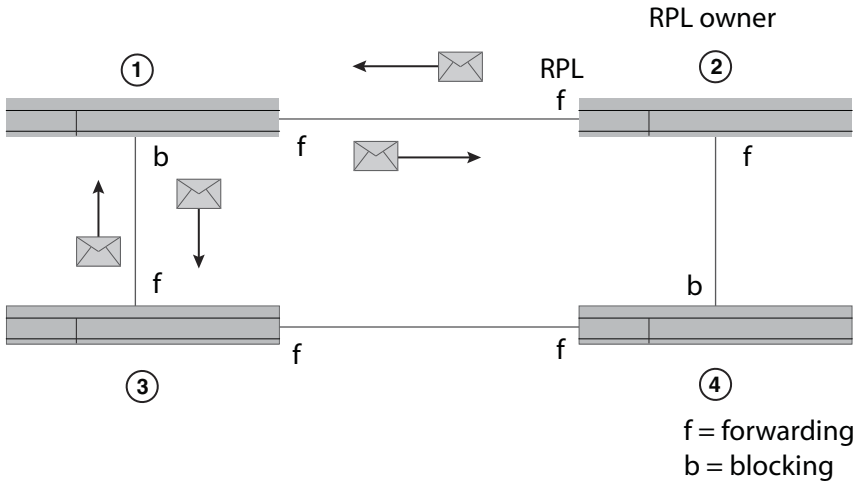


Figure 77 shows the next sequence of events. The actions of each ERN are:

- ERN 1 terminates R-APS received on the blocked port, unblocks the non-failed port, stops transmitting R-APS (NR) messages, and enters the Pending state.
- ERN 2 takes no action.
- ERN 3 takes no action.
- ERN 4 stays in the Pending state and transmits R-APS (NR) messages.

FIGURE 77 Initializing an ERP topology - III

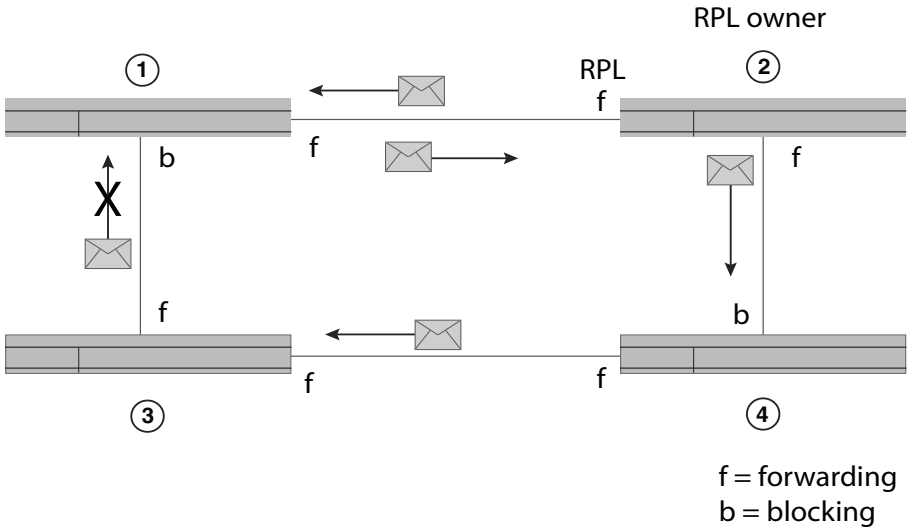


Figure 78 shows the next sequence of events. The actions of each ERN are:

- ERN 1, from the Pending state, unblocks the left interface, stops sending R-APS (NR) and stays in the Pending state. Now both interfaces are in the forwarding state.
- ERN 2 takes no action.
- ERN 3 takes no action.
- ERN 4 stays in the Pending state and transmits R-APS (NR) messages. The left interface is blocked, and the right interface is in the forwarding state.

FIGURE 78 Initializing an ERP topology - IV

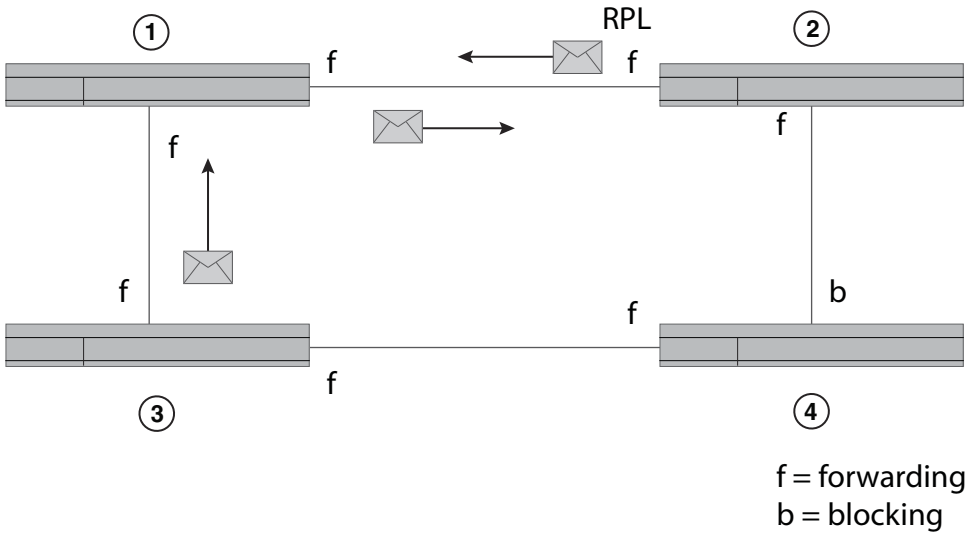


Figure 79 on page 495 shows the next sequence of events. Next ERN 2 initializes. The actions of each ERN are:

- ERN 1 stays in the Pending state.
- ERN 2 (RPL owner), from the Init state, stops the guard timer, stops the WTB timer, blocks the RPL, unblocks the non-RPL port, enters the Pending state, transmits R-APS (NR) messages, and starts the WTR timer.
- ERN 3 takes no action.

- ERN 4 stays in the Pending state and transmits R-APS (NR) messages. The left interface is blocked.

FIGURE 79 Initializing an ERP topology - V

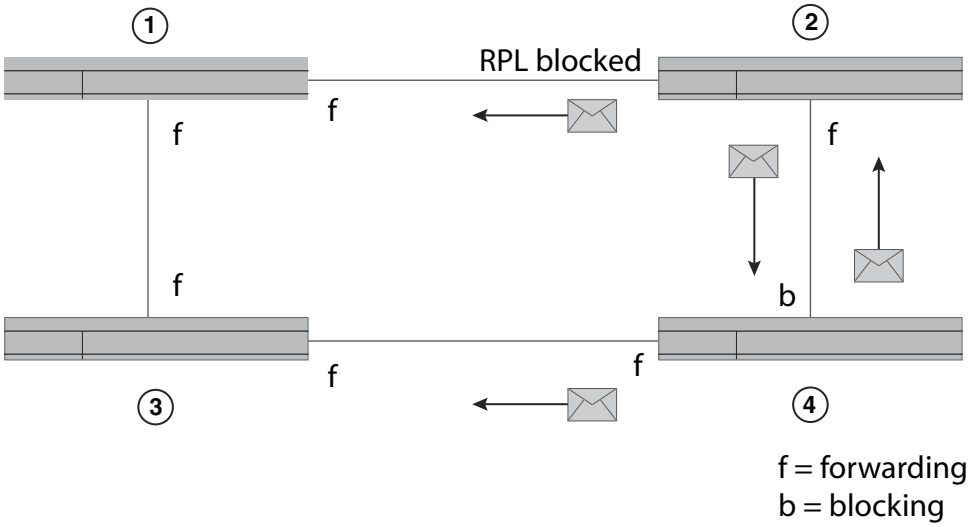
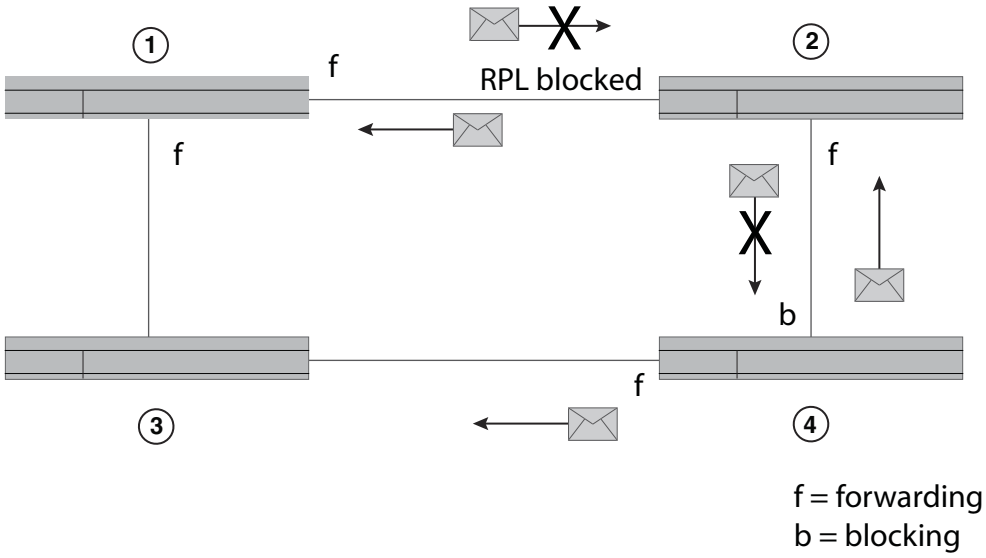


Figure 80 shows the next sequence of events. The actions of each ERN are:

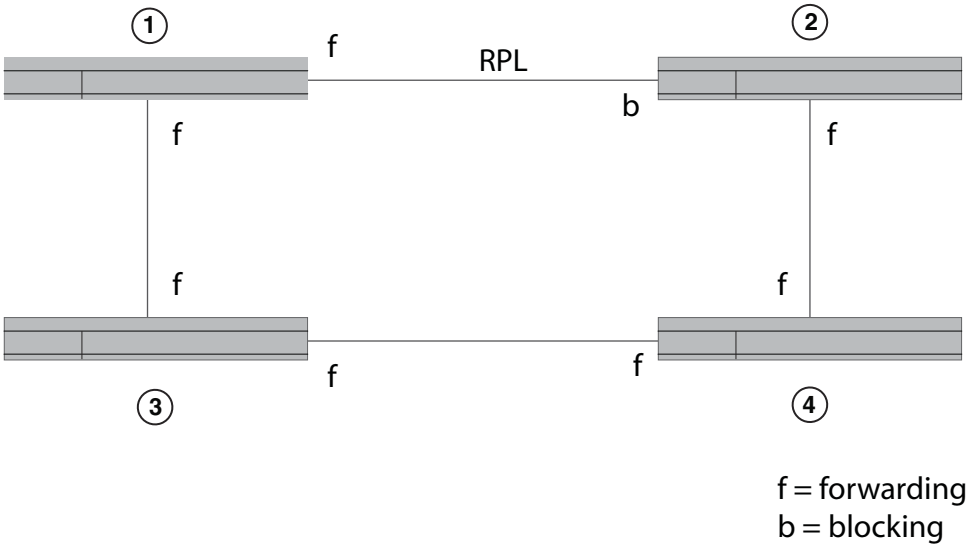
- After the WTB timer expires, ERN 2 (RPL owner in the Pending state) transmits R-APS (NR, RB), and then ERN 2 enters the Idle state.
- ERN 1, still in the Pending state, forwards R-APS (NR, RB) and enters the Idle state.
- ERN 3 takes no action.
- ERN 4 from the Pending state and stops transmitting R-APS (NR).

FIGURE 80 Initializing an ERP topology - VI



Lastly, ERNs 1, 2, and 3 are in the Idle state, and ERN 4 changes the blocking port to the forwarding state. All ERNs remain in the Idle state. See Figure 81.

FIGURE 81 Initializing an ERP topology - VII



Signal fail

Signal fail and signal fail recovery provide the mechanism to repair the ring to preserve connectivity among customer networks.

ERP guarantees that although physically the topology is a ring, logically it is loop-free. One link, called the Ring Protection Link (RPL), is blocked to traffic. When a non-RPL link fails in the ring, the signal failure mechanism triggers and causes the RPL to become forwarding. Later, signal fail recovery can occur to restore the ring to the original setup.

Convergence time is the total time that it takes for the RPL owner to receive the R-APS (NR) message and block the RPL port until the ERN with the failed link receives notice and unblocks the failed link.

Figure 82 shows a simple Ethernet ring topology before a failure. This diagram shows dual-end blocking enabled (thick line) between ERNs one (RPL node) and 6 (RPL owner). ERNs 3, 2, 4, and 5 are non-RPL nodes.

FIGURE 82 ERP topology

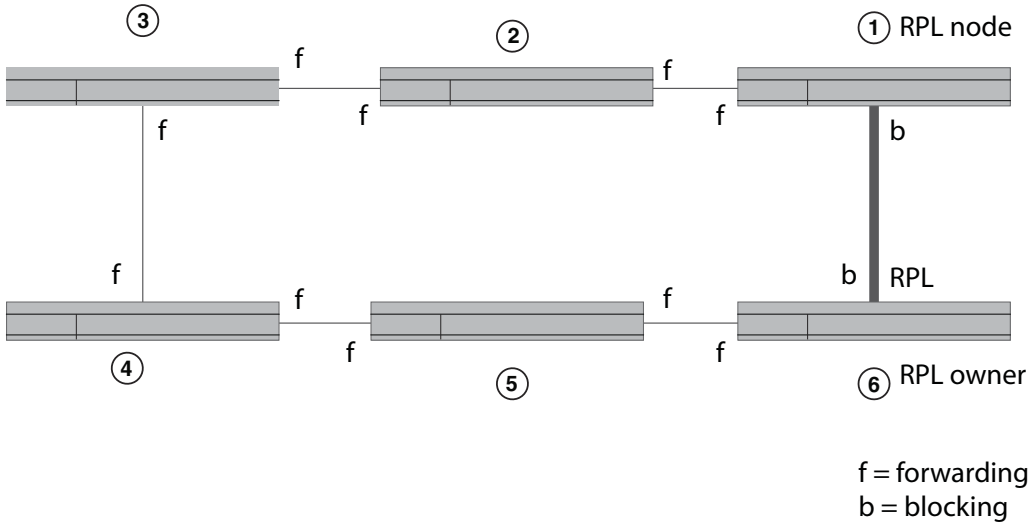
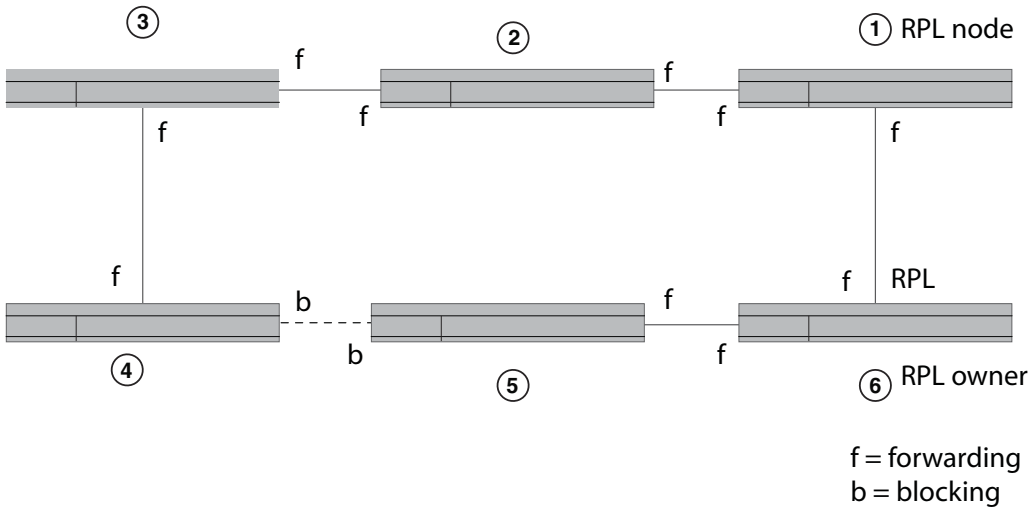


Figure 83 shows the same Ethernet ring topology after a failure at the forwarding port of ERN 4 when a signal fail triggered, and ring protection was needed. ERN 6 unblocked the RPL port and the RPL node changed the blocking port to the forwarding state.

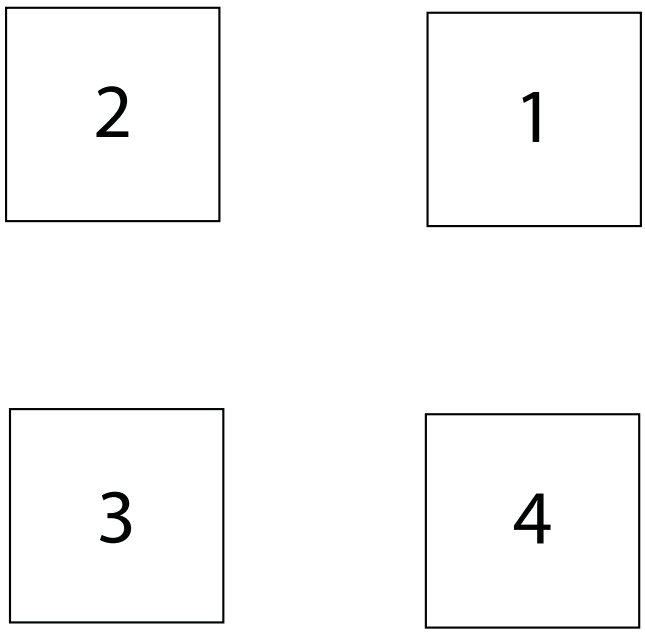
FIGURE 83 ERP topology in a Protected state



Manual switch

In the absence of a failure, an operator-initiated manual switch (MS) moves the blocking role of the RPL by blocking a different ring link and initiates the node sending a R-APS (MS) to inform the RPL owner to unblock the RPL. This can occur if no higher priority request exists in the ring. See Figure 84. The thick line between ERNs 1 and 2 indicate that dual-end blocking is enabled.

FIGURE 84 Manual Switch example



The node, which receives the R-APS (MS), forwards it to the adjacent nodes. If the receiving node is already in the Idle or Pending state, it unblocks the non-failed port and stops transmitting R-APS messages. Only one MS can exist in the topology at any time. An MS condition has to be manually cleared with the **no** command.

NOTE
If any ERN is in an FS state or in a protected state through an SF event and an operator tries to configure an MS, the ERN will reject the request.

When a manual switch is cleared by an operator on the same node on which the MS is configured, the node keeps the port in a blocking state, sends out a R-APS (NR) to the adjacent node, and starts the guard timer. Other nodes that receive the R-APS (NR) forward the message. When the RPL owner receives this message, then the RPL owner starts the WTR timer. When the WTR timer expires, the RPL owner sends out a R-APS (NR, RB), blocks the RPL, and flushes the FDB. Other nodes in the topology that receive the R-APS (NR, RB) unblock any non-failed port and flush the FDB.

Figure 84 shows a manual switch on ERN 3, which is a non-RPL node. In order to clear the MS condition, the operator must enter the manual switch command from ERN 3. The sequence of messages and actions is as shown in Figure 85 on page 499.

FIGURE 85 MS on Non-RPL node

Non-RPL node with error (ERN 3)	RPL owner (ERN 1) and RPL node (ERN2)	Other Non-RPL node (ERN 4)
From the Idle state, ERN3: 1 Blocks the MS port 2 Sends the RAPS (MS) 3 Flushes the FDB 4 Enters the manual switch (MS) state		
		From the Idle state, ERN 4: 1 Forward R-APS (MS) 2 Flush the FDB 3 Enter the MS state
	From the Idle state, ERN 1: 1 Forwards R-APS (MS) 2 Unblocks the RPL 3 Flushes the FDB 4 Enters the MS state	

After the manual switch is triggered, the operator can clear it with the **no** command and MS recovery will begin. Figure 86 shows the sequence of events during the MS recovery process.

FIGURE 86 MS recovery process

Non-RPL node with error (ERN 3)	RPL owner (ERN 1)	RPL node (ERN2) with dual-end blocking enabled	Non-RPL node (ERN 4)
From the MS state, ERN 3: 1 Stops sending R-APS (MS) 2 Sends R-APS (NR) 3 Continues to block the port 4 Enters the Pending state			
	From the MS state, ERN 1: 1 Receives the R-APS (NR) 2 Starts the WTB timer 3 Forwards the R-APS (NR) 4 Enters the Pending state 5 After the WTB timer expires, blocks the RPL 6 Flushes the FDB 7 Sends R-APS (NR, RB) 8 Enters the Idle state	From the MS state, ERN 2: 1 Receives the R-APS (NR) 2 Forwards the R-APS (NR) 3 Enters the Pending state	From the MS state, ERN 2: 1 Receives the R-APS (NR) 2 Forwards the R-APS (NR) 3 Enters the Pending state
From the Pending state, ERN 3: 5. Receives the R-APS (NR, RB) and unblocks the blocking port 6. Forwards the R-APS (NR, RB) 7. Flushes the FDB 8. Enters the Idle state		From the Pending state, ERN 2: 4. Blocks the RPL 5. Forwards the R-APS (NR, RB) 6. Flushes the FDB 7. Enters the Idle state	From the Pending state, ERN 4: 4. Forwards the R-APS (NR, RB) 5. Flushes the FDB 6. Enters the Idle state

Forced switch

Forced switch (FS) is an operator-initiated mechanism that moves the blocking role of the RPL to a different ring link followed by unblocking the RPL, even if one or more failed links exist in the ring.

The node configured to initiate an FS blocks the port and sends out a R-APS (FS) to inform other nodes to unblock any blocked ports (including failed ones) as long as no other local request with higher priority exists. The RPL owner unblocks the RPL and flushes the FDB.

Any node accepting a R-APS (FS) message stops transmitting R-APS messages.

Multiple FS instances can be configured in the topology even when the topology is in the same segment where an FS is being cleared by **no** command. When an operator clears an FS on the same node where an FS is configured, this node keeps the port in the blocking state, sends out a R-APS (NR) to adjacent nodes, and starts the guard timer. Other nodes that receive the R-APS (NR) forward the message. When the RPL owner receives this message, the RPL owner starts the WTB timer. When the WTB timer expires, the RPL owner sends out a R-APS (NR, RB), blocks the RPL, and flushes the FDB. Other nodes in the topology that receive the R-APS (NR, RB) unblock any non-failed port and flush the FDB.

An FS request can be accepted no matter what state the topology is in. Since the local FS and R-APS (FS) are higher priority than SF; an SF occur i ng later than FS will not trigger the SF process. In addition, because the local FS and R-APS (FS) are higher priority than SF, when a node receives a R-APS (FS) without any local higher priority event, it will unblock any blocked port. The node with the failed link also unblocks the blocked port; but because the link has failed, the topology is broken into segments.

Since the local FS and R-APS (FS) are higher priority than a local SF clear when the link failure is removed without any local higher priority event, the nodes with the recovering link do not trigger SF recovery.

After the operator clears the FS condition on the node, the node starts the guard timer and sends out a R-APS (NR). When the RPL owner receives a R-APS(NR), it stops the WTB timer and starts the guard timer. The RPL owner blocks the RPL and sends out a R-APS (NR, RB). Any node receiving a R-APS (NR, RB) unblocks the non-failed blocked port. If the guard timer is still running on the node with previous FS, this node ignores R-APS messages until the guard timer expires. The topology is again broken into segments. After this node processes the R-APS (NR, RB), however, it unblocks the blocked node; and the topology is in a loop free state and in one segment.

Figure 87 shows a port failure on ERN 4.

FIGURE 87 Single forced switch scenario

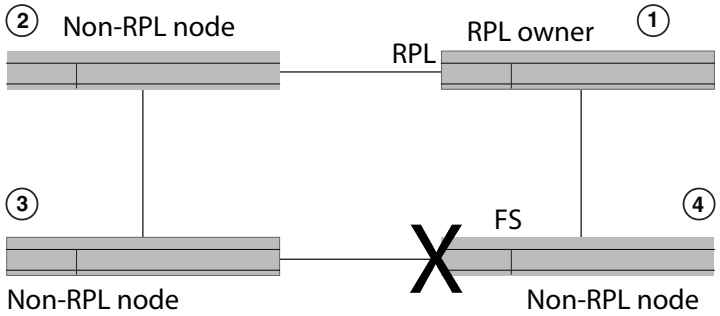


Figure 88 shows the sequential order of events triggered as a result of an operator-initiated forced switch command entered from ERN 4.

FIGURE 88 Single FS process–operator entered the forced switch command from ERN 4

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
Idle	Idle	Idle	From the Idle state, ERN 4: 1 Processes the Forced Switch command 2 Blocks the requested port 3 Transmits R-APS (FS) 4 Unblocks the non-requested port 5 Flushes the FDB 6 Enters the Forced Switch (FS) state
From the Idle state, ERN 1: 1 Unblocks the RPL 2 Flushes the FDB for first time 3 Forwards R-APS(FS) 4 Enters the FS state	From the Idle state, ERN 2: 1 Unblocks the port 2 Flushes the FDB for the first time 3 Forwards R-APS(FS) 4 Enters the FS state	From the Idle state, ERN 3: 1 Unblocks the port 2 Flushes the FDB for the first time 3 Forwards R-APS(FS) 4 Enters the FS state	
From the FS state, ERN 1 forwards R-APS	From the FS state, ERN 2 forwards R-APS	From the FS state, ERN 3 forwards R-APS	From the FS state, ERN 4: 7. Transmits R-APS(FS) 8. Terminates the received R-APS on the blocking port 9. Terminates its own R-APS(FS)
All ERNs remain in FS state.			

Next, the operator enters the **no** command to clear the forced switch. For this example, the operator initiated the forced switch from ERN 4 and must clear it from ERN 4. Figure 89 on page 502 shows the forced switch recovery process in sequential order.

FIGURE 89 FS clear process

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
			From the FS state, ERN 4: 1 Starts the guard timer 2 Stops transmitting R-APS(FS 3 Transmits R-APS(NR) 4 Keeps blocking the port 5 Enters Pending state
From FS state, ERN 1: 1 Forwards R-APS 2 Starts the guard timer 3 Starts the WTB timer 4 Enters Pending state			
	From FS state, ERN 2: 1 Forwards R-APS 2 Starts the guard timer 3 Enters the Pending state	From FS state, ERN 3: 1 Forwards R-APS 2 Starts the guard timer 3 Enters the Pending state	
After the WTB timer expires, from the Pending state ERN 1: 5. Blocks the RPL port 6. Transmits R-APS(NR,RB) 7. Unblocks the non-RPL port 8. Flushes the FDB 9. Enters the Idle state			
	From the Pending state, ERN 2: 4. Flushes the FDB 5. Forwards R-APS(NR,RB) 6. Enters the Idle state	From the Pending state, ERN 3: 4. Stops transmitting R-APS 5. Unblocks ports 6. Flushes the FDB 7. Forwards R-APS(NR,RB) Enters the idle state	From the Pending state, ERN 4: 6. Stops transmitting R-APS 7. Unblocks ports 8. Flushes the FDB 9. Forwards R-APS(NR,RB) 10. Enters the Idle state
From the idle state, ERN 1: 10. Receives its own R-APS(NR,RB) 11. Stops transmitting R-APS 12. Remains in the Idle state			

Double Forced Switch

A local FS is of a higher priority than a received R-APS (FS); therefore, the local FS request blocks the port even when the node receives a R-APS(FS) from another FS request of another node.

After the first FS clears, the node starts the guard timer and sends out a R-APS (NR). The adjacent nodes of the first cleared FS node will not process or forward the R-APS (NR) because they are still receiving R-APS (FS) from the second FS node. When the first FS node receives R-APS (FS) from the second FS nodes, it unblocks any blocked port and stops transmitting any lower priority R-APS messages. At this point, the topology follows the single FS process, as previously described.

Dual-end blocking

Dual-end blocking is a user configurable feature to directly conserve bandwidth of the RPL and indirectly conserve processing power of the RPL owner. When you configure a node in a major ring adjacent to the RPL owner to be an RPL node with dual-end blocking enabled, data traffic and R-APS messages will not be forwarded to the blocked port of the RPL owner.

When a failure occurs in the ring and the RPL node (not the RPL owner) receives a R-APS (of type SF, FS, or MS), the RPL node unblocks the configured dual-end blocked port. When the RPL node receives a R-APS (NR, RB), it reblocks the originally configured dual-end blocked port. To configure dual-end blocking you need to configure the RPL and dual-end blocking on both the RPL owner and the adjacent peer (RPL node).

Non-revertive mode

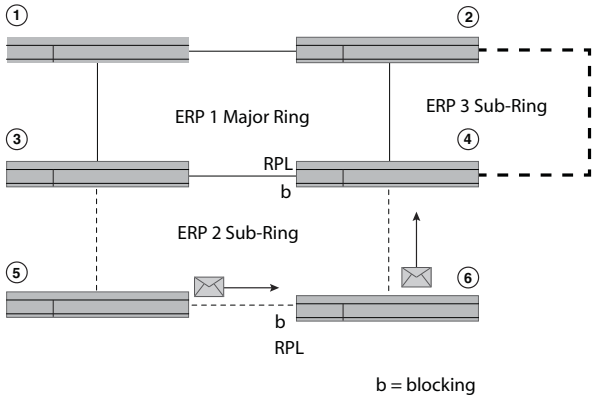
In non-revertive mode, the traffic channel is allowed to use the RPL, if it is not failed, after a switch condition clears. In the recovery from a Protection state, the RPL owner generates no response regarding the reception of NR messages. When other healthy nodes receive the NR message, there is no action in response to the message. After the operator issues a **no** command for non-revertive mode at the RPL owner, the non-revertive operation is cleared, WTB or WTR timer starts, as appropriate, and the RPL owner blocks its port to the RPL and transmits a R-APS (NR, RB) message. Upon receiving the R-APS (NR, RB), any blocking node should unblock its non-failed port.

Interconnected rings

Interconnected Rings consist of one major ring and one or more sub-rings with shared physical links. The ring links between the interconnection nodes are controlled and protected by the ERP ring to which they belong. A sub-ring is similar to the major ring in that each sub-ring has an RPL and an RPL owner. The RPL owner can be configured in any node belonging to the ring.

The dotted lines in Figure 90 on page 504 show two of the many potential sub-rings that you can configure.

FIGURE 90 Interconnected rings with major and sub rings shown



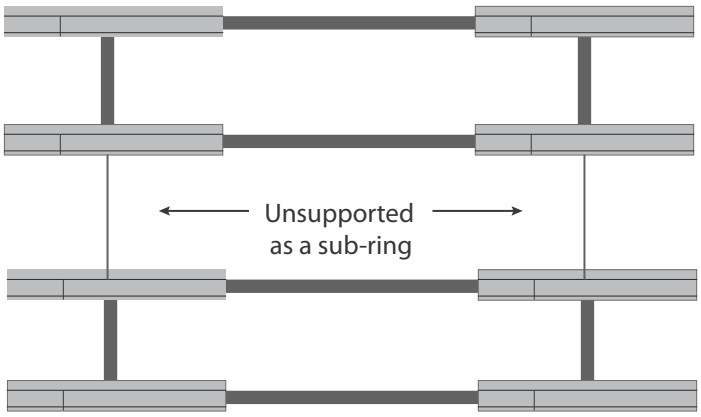
When a sub-ring initializes, each ERN in the non-closed ERP sends out a R-APS (NR). After the RPL owner receives a R-APS (NR), it blocks the RPL; and the RPL owner sends out a R-APS (NR, RB). The shared link remains blocked even if the shared link has a SF error. The blocking state in ERP means the R-APS channel is blocked at the same port where the traffic channel is blocked, except on sub-rings without use of R-APS virtual channel.

Virtual channel support is not available in release 5.1.0. A sub-ring in segments interconnecting major rings is not supported. Figure 91 on page 504 shows a major ring and two segments not supported as a sub-ring.

Blocking prevents R-APS messages received at one ring port from being forwarded to the other ring port; it does not prevent the R-APS messages locally generated at the ERP control process from being transmitted over both ring ports, and it also allows R-APS messages received at each port to be delivered to the ERP control process.

Each ERN in a major ring terminates R-APS messages received on a blocking port and does not forward the message if the port is in a blocking state. Each ERN in a sub-ring, however, still forwards the R-APS messages received on a blocking port.

FIGURE 91 Unsupported sub-ring in segments



FBD flush optimization

The FDB stores the node ID and BPR sent in the R-APS messages. When an ERN receives a new R-APS message, it compares the received node ID and BPR to the node ID and BPR in its memory. If the pair vary from the previously stored pair, the ERN deletes the previous pair and stores the new pair. The device then triggers a FDB flush unless the DNF (No Not Flush) is set in the message.

FDB optimization is achieved with the following features:

- Non-revertive mode alleviates the need to flush the FDB after a link failure with link protection (SF) condition
- Dual-end blocking decreases attempted messages and traffic to the RPL blocking port
- Interconnected ring support to decrease the latency for messaging
- Do Not Flush (DNF) messages

Configuring ERP

To configure and initialize ERP using only APS you must set up one RPL owner and one or more Non-RPL nodes. The minimum configuration tasks are listed in this section.

Before configuring ERP, however, you must have already configured a VLAN and ports.

You must perform the following minimum configuration tasks for the RPL owner:

- Configure an ERP instance
- Set the left and right interfaces
- Set the role as owner
- Set the RPL
- Enable the configuration

You must perform the following minimum configuration tasks for each non-RPL node:

- Configure an ERP instance
- Set the left and right interfaces
- Enable the configuration

Sample configuration

The following example is of an ERP configuration consisting of four devices: an RPL owner, an RPL node, and two non-RPL nodes.

NOTE
Before configuring any ERP settings, configure the VLAN and ports.

Device 1 RPL owner

NOTE
Optionally, you can configure the non-revertive mode feature. This setting can only be set on the RPL owner.

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#rpl vlan 2 e1/2
(config-erp-1)#rpl-owner
(config-erp-1)#enable
```

Device 2 RPL node

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#rpl vlan 2 e 1/1
(config-erp-1)#enable
```

Device 3 Non-RPL node

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#enable
```

Device 4 Non-RPL node

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#enable
```

Configuring ERP with IEEE 802.1ag

To configure and initialize ERP using APS and IEEE 802.1ag you must set up one RPL owner and one or more Non-RPL nodes. Other nonparticipating switches can exist in the ring.

You must perform the following minimum configuration tasks for the RPL owner:

- Configure an ERP instance
- Set the left and right interfaces
- Set the role as owner
- Set the RPL
- Enable the configuration

You must perform the following minimum configuration tasks for each non-RPL node:

- Configure an ERP instance
- Set the left and right interfaces

- Configure the maintenance entity group end points (MEP) from each ERN, which can have a role of RPL owner or non-RPL node, adjacent to switches not participating in the ERP configuration
- Enable the configuration

ERP commands

This section lists ERP configuration commands.

Assigning ERP IDs

You must assign an ERP ID. This ID number is used to:

- Filter and clear statistics associated with a particular ERP ID
- Delete the non-revertive mode in the case of an RPL owner
- Clear WTR and WTB timers

The *erp_id* value is a number from 1 to 255.

Syntax: erp <*erp_id*>

For example, to assign the number 10 to the ERP, enter:

```
(config)# erp 10
```

Naming an Ethernet Ring Node

From within the ERP configuration shell, you can optionally name an ERN with a meaningful name. The name must be 31 alphanumeric characters or fewer; and the name can use the “underscore” and “dash” special characters.

Syntax: [**no**] **name** <*erp_name*>

For example, to assign the name “to_dell1” to an ERN with ID number 10, enter:

```
(config)# erp 10
(config-erp-10)# name “to_dell1”
```

Use the **no** command to remove the name.

Configuring the default MAC ID

You can configure the MAC ID. The router appends this ID number to the end of the permanent portion of the ERP MAC address (01-19-A7-00-00- <01 or ERP ID>) in R-APS messages. By default 01-19-A7-00-00-01 is used as the dst MAC, which is always used by Version 1 of ITU-T 8032. If Version 2 is configured, then the **raps-default-mac** command can be negated by entering the **no raps-default-mac** command. The configured ERP ID will appear as the last 8-bit number in the destination MAC.

For more information about feature support for version 1 and 2, see “Setting the ITU-T G.8032 version number” on page 512.

Syntax: [no] raps-default-mac

Enabling the ERP configuration

You must apply the **enable** command to activate an ERP configuration. You can use the **no** command to disable the configuration.

Within an interconnected ring topology, in the major ring, you must first configure two interfaces. In a sub-ring, at least one interface must first be configured before enabling the ERP instance.

Syntax: [no] enable

Example of a non-RPL node configuration in a major ring:

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#enable
```

Configuring interfaces

Each ERN in a major ring must have explicitly defined left and right interfaces so that ERP can function properly. ERNs in a sub-ring must have at least one interface defined so that ERP can function properly.

For proper operation you must configure the interfaces following the same manner on each ERN, such as left/right, left/right, and so on.

Syntax: [no] left-interface [vlan <*vlan-id*> e <*slot/number*>]

Syntax: [no] right-interface [vlan <*vlan-id*> e <*slot/number*>]

Use the **no** command to remove the configuration of each interface.

Assigning the RPL owner role and setting the RPL

Each ring needs to have one RPL owner for each ring. The RPL owner's role is to block traffic on one port when no failure exists in the ring. The blocked port will be the left interface that you initially configured. After configuring the ERN to be the RPL owner, you next must set the RPL. To set the RPL you need to specify the VLAN and Ethernet slot and port.

NOTE
When you assign the role of RPL owner, you must also configure the RPL.

Syntax: [no] rpl-owner

Syntax: [no] rpl [vlan <*vlan-id*> e <*slot/number*>]

Enabling sub-rings for multi-ring and ladder topologies

In multi-ring and ladder topologies, you can enable the multi-ring feature.

Interconnected rings consists of one major ring and at least one sub-ring within the same VLAN. A sub-ring is not a complete ring. Nodes within a sub-ring can be configured as a one-arm ring. Each sub-ring must have its own RPL owner and RPL ports as appropriate.

RPL ports and the RPL owner also need to be configured in a sub-ring. All ERP features are available in both major and sub-rings.

R-APS PDUs only flow in the nodes with same ring ID. The R-APS PDU can be forwarded through the port in sub-ring blocking state.

Syntax: [no] **sub-ring**

Use the **no** command to delete the sub-ring support.

If you have six nodes you can put them in one ring. The latency time for packet transport, however, increases in big topologies even within the same VLAN, so it is better to separate them out.

Configuring non-revertive mode

After the Ethernet Ring enters a protected state, if you do not want the topology to return to the original state you can use the **non-revertive-mode** command to keep it in the new state. Enter this command on the RPL owner only, and then enter the **enable** command.

Syntax: [no] **non-revertive-mode**

Use the **no** keyword to remove the non-revertive mode setting.

Configuring and clearing a forced switch

An operator can use the forced switch (FS) mechanism when no errors, a single error, or multiple errors are present in the topology. You can enter this command multiple times. You need to explicitly specify the VLAN and Ethernet slot and port.

Syntax: [no] **forced-switch** [**vlan** <*vlan*> **e** <*slot/port*>]

Use the **no** command to remove the forced switch mechanism.

Configuring and clearing a manual switch

Manual switch (MS) is an operator-initiated process that manually blocks a desired port in a ring. You need to explicitly specify the VLAN, Ethernet slot, and port from the desired device.

Syntax: [no] **manual-switch** [**vlan** <*vlan*> **e** <*slot/port*>]

Use the **no** command to remove the manual switch mechanism.

Configuring dual-end blocking

You can configure dual-end blocking to optimize your ERP configuration. The RPL node must be adjacent to the RPL owner.

When you configure the RPL on an ERN that is adjacent to the RPL owner, you are enabling the dual-end blocking feature and changing the ERN's role to that of RPL node. You configure the RPL node with the **rpl** command. Before configuring dual-end blocking, you must verify that the RPL node is actually the correct peer and obtain the RPL link settings; an incorrect setting will cause incorrect port blocking.

NOTE
The RPL node must be a peer of the RPL owner, and the RPL must be configured on this peer; otherwise, the device will perform incorrect port blocking behavior.

Syntax: [**no**] **rpl** [**vlan** <*vlan-id*> **e** <*slot/number*>]

Configuring the guard timer

The guard timer prevents ERNs from acting upon outdated R-APS messages and prevents the possibility of forming a closed loop. The guard timer enforces a period during which an ERP topology ignores received R-APS.

This timer period should always be greater than the maximum expected forwarding delay in which an R-APS message traverses the entire ring. The longer the period of the guard timer, the longer an ERN is unaware of new or existing relevant requests transmitted from other ERN and, therefore, unable to react to them.

The guard timer is used in every ERN, once a guard timer is started, it expires by itself. While the guard timer is running, any received R-APS request/state and Status information is blocked and not forwarded to the priority logic. When the guard timer is not running, the R-APS request/state and status information is forwarded unchanged.

NOTE
The Release 5.1.0 implementation of the guard timer differs from the ITU-T G.8032 document. The standard defines the guard timer period as configurable in 10 ms increments from 10 ms to 2000 ms (2 seconds) with a default value of 500 ms.

The guard timer is activated when an ERN receives an indication that a local switching request, such as a clear signal fail, manual switch, or forced switch, is cleared.

The guard timer can be configured in 100ms increments from 1200ms to 4000ms (4 seconds); the default value is 1500ms (1.5 seconds). The guard timer cannot be stopped manually.

Syntax: **guard-time** <*time-value*>

Configuring and clearing the wait to restore timer

For SF recovery situations, you can configure the wait to restore (WTR) timer on the RPL owner to prevent frequent operation of the protection switching due to the detection of intermittent signal failures. When recovering from a Signal Failure, the WTR timer must be long enough to allow the recovering network to become stable.

This WTR timer is activated on the RPL Owner Node. When the relevant delay timer expires, the RPL owner initiates the reversion process by transmitting an R-APS (NR, RB) message. The WTR timer is deactivated when any higher priority request preempts this timer. The WTR timers may be started and stopped. A request to start running the WTR timer does not restart the WTR timer. A request to stop the WTR timer stops the WTR timer and resets its value. The Clear command can be used to stop the WTR timer. While WTR timer is running, the WTR running signal is continuously generated. After the WTR timer expires, the WTR running signal is stopped, and the WTR Expires signal is generated. When the WTR timer is stopped by the clear command, the WTR Expires signal is not generated.

When configured, the RPL owner waits until the timer expires before transmitting the R-APS(NR,RB) message to initiate the reversion process. While the timer is in effect, the WTR running signal is continuously generated. You can configure the WTR timer in 1 minute increments from 1 to 12 minutes; the default value is 5 minutes.

This timer can be stopped by issuing the **clear erp** <*erp_id*> **wtr-timer** command.

Syntax: wtr-time <*time-value*>

Testing the WTR timer

You can enter the **fast-wtr-time** command to test your configuration. Instead of having to wait 5 minutes for the timer to expire, you wait 5 seconds. This command changes the timer's unit of measure from minutes to seconds.

Syntax: [**no**] **fast-wtr-time**

Use the **no** command to return the unit of measure to minutes.

Configuring and clearing the WTB timer

The WTB timer ensures that clearing of a single FS command does not trigger the reblocking of the RPL when multiple FS situations co-exist in an Ethernet Ring. When recovering from an MS or FS command, the delay timer must be long enough to receive any latent remote FS or MS.

While it is running, the WTB running signal is continuously generated. The WTB timer is 5000ms (5 seconds) longer than the guard timer. You can configure this timer in 100 ms increments from 5100ms to 7000ms (7 seconds); the default value is 5500ms.

The WTB timer can be stopped through the CLI by entering the **clear erp** <*erp_id*> **wtb-timer** command.

Syntax: wtb-time <*time-value*>

Configuring a hold-off timer

The hold-off timer is used in each ERN to prevent unnecessary Signal Fail events due to port flapping. If you configure a non-zero hold-off timer value, when a link error occurs, the event will not be reported immediately. When the hold-off timer expires, ERP checks if the error still exists.

The hold-off timer is used in every ERN. When a new defect occurs (new SF), this event will not be reported immediately to trigger protection switching if the provisioned hold-off timer value is non-zero. Instead, the hold-off timer is started. When the hold-off timer expires, the trail that started the timer is checked as to whether a defect still exists. If one does exist, that defect is reported and protection switching is triggered.

You can configure the hold-off timer in 100ms increments from 0 to 10,000 ms (10 seconds); the default value is 0 ms. The hold-off timer value cannot be stopped through the CLI.

Syntax: holdoff-time <*time-value*>

Configuring the message interval time

The message interval time of R-APS messages continuously sent within an ERP ring can be configured. You can configure the interval in 100ms increments from 100ms to 5000ms (5 seconds); the default value is 5000ms.

Syntax: message-interval <*time-value*>

Configuring IEEE 802.1ag support

You can enable IEEE 802.1ag support by entering the **dot1ag-compliance** command. By enabling this protocol, ERNs with non-ERN switches between them can be notified when the domain goes down. You need to enter the domain name, MA name, and MEP value of the domain with which to associate this ERP.

Syntax: [**no**] **dot1ag-compliance domain-name** <*name*> **ma-name** <*name*> **mep** <*mep_value*>

Setting the ITU-T G.8032 version number

You can configure the ERP configuration to use G.8032 version 1 or 2. The default value is version 2. Table 92 lists the feature and MAC ID differences between versions 1 and 2.

NOTE
The ERP **version** command does not have a shortened form. You must enter the complete command.

TABLE 92 Feature differences between G.8032 version 1 and 2

Version	Supported ERP features	Treatment of MAC ID
1	• Signal Fail • Signal Fail recovery	Always uses 01:19:A7:00:00:01 as the ERP ID in R-APS messages
2	• Signal Fail • Signal Fail recovery • Manual Switch • Forced Switch • Non-revertive • Interconnected rings • RPL configuration on non-RPL owner	Allows use of the ERP ID for the last two bytes of the MAC ID (01:19:A7:00:00:erp-id)

Syntax: version < *version_number*>

You can view the version by entering the **show erp** command. The version appears on the top line directly after the ERP ID.

Viewing ERP operational status and clearing ERP statistics

You can view operational status and statistics and clear statistics for all links or particular links.

Viewing ERP operational status and statistics

To view ERP statistics, enter the following command on the RPL owner:

Syntax: show erp [<*enter*> | <*erp_id*>]

To view ERP information for all links, enter **show erp** followed by pressing the Enter key (carriage return). To view statistics for a particular link, enter the ERP ID after the command.

Example output:

```
NetIron#show erp 7
ERP 7(version 2)- VLAN 504
================================================================
Erp    ID     Status     Oper       Node       Topo
                         state      role       group
       1      enabled    Idle       rpl-owner  -

Ring type  WTR        WTB        Guard      Holdoff    Msg
           time(min)  time(ms)   time(ms)   time(ms)   intv(ms)
Major-ring 5          7000       2000       0          1000
```

```
I/F    Port   ERP port state    Interface status     Interface type
L      1/12   blocking          normal               rpl
R      1/11   forwarding        normal            non-rpl

RAPS sent     RAPS rcvd     RAPS dropped  RAPS ignored     Oper state changes
3             3             0             0                0
```

Table 93 summarizes the table fields and their meanings.

TABLE 93 Summary of CLI output for **show erp** command

This field...	Displays...
ERP id	The ERP ID is the number that was configured at setup. The ERN appends this number to the permanent portion of the MAC address (01-19-A7-00-00) used for ERP.
Status	Enabled or disabled
Operational state	Init, Idle, Protection, Manual Switch, Forced Switch or Pending
Node role	rpl-owner, non-rpl-node or rpl-node
Topology group	<topology group id> or "- " (- means N/A)
Ring type	Major-ring or Sub-ring
Timers	Configuration value for each timer
Interfaces (I/F)	L (left) or R (right)
Port	<slot/port>
ERP port state:	disabled, blocking, forwarding
Interface status:	normal, signal-fail, manual-switch or forced-switch
Interface type:	rpl or non-rpl
RAPS sent:	RAPS sent by MP (self generated)
RAPS rcvd:	RAPS received by MP
RAPS dropped:	RAPS dropped by MP
RAPS ignored:	RAPS ignored (for example, the guard-timer, or non regular type)

Clearing ERP statistics

You can clear ERP statistics by entering the **clear erp** <erp_id> **statistics** command to clear the statistics of one erp instance. You can clear all ERP statistics by entering the **clear erp statistics** command to clear the statistics of all erp instances.

Syntax: **clear erp** <*erp_id*> **statistics**

clear erp statistics

Virtual Switch Redundancy Protocol (VSRP) 15

The following Virtual Switch Redundancy Protocol (VSRP) features are supported by the .

- Virtual Switch Redundancy Protocol (VSRP)
- VSRP 2
- Layer 2 Redundancy
- Layer 3 VSRP
- MAC Address Failover on VSRP-Aware Devices
- VSRP Fast Start
- VSRP Slow Start
- VSRP and Foundry MRP Signaling

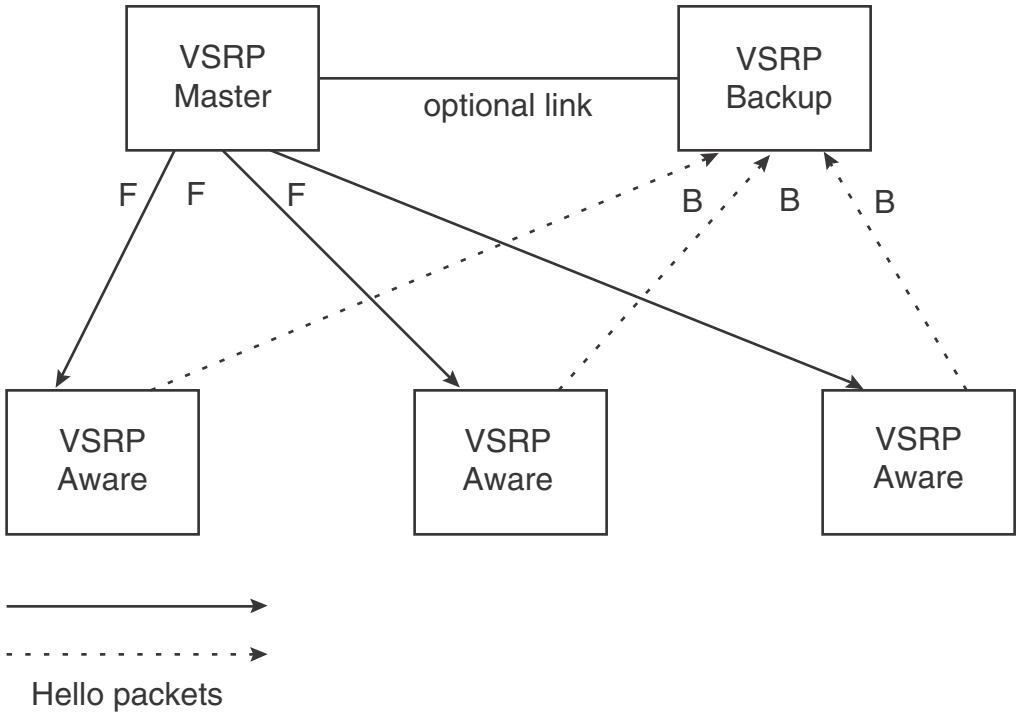
VSRP is a proprietary protocol that provides redundancy and sub-second failover in Layer 2 mesh topologies. Based on the Virtual Router Redundancy Protocol Extended (VRRPE), VSRP provides one or more backups for the PowerConnect. If the active PowerConnect becomes unavailable, one of the backups takes over as the active device and continues forwarding traffic for the network. You can use VSRP for the PowerConnect.

VSRP is a Dell proprietary protocol that provides redundancy and sub-second failover in Layer 2 and Layer 3 mesh topologies. Based on the Dell's proprietary Virtual Router Redundancy Protocol Extended (VRRPE), VSRP provides one or more backups for the device. If the active device becomes unavailable, one of the backups takes over as the active device and continues forwarding traffic for the network.

Layer 2 and Layer 3 share the same VSRP configuration information.

Figure 92 shows a VSRP configuration.

FIGURE 92 VSRP mesh – redundant paths for Layer 2 traffic



In this example, two PowerConnect devices are configured as redundant paths for VRID 1. On each PowerConnect, a Virtual Router ID (VRID) is configured on a port-based VLAN. Since VSRP is primarily a Layer 2 redundancy protocol, the VRID applies to the entire VLAN. However, you can selectively remove individual ports from the VRID if needed.

Following Master election (described below), one of the devices becomes the Master for the VRID and sets the state of all the VLAN's ports to Forwarding. The other device is a Backup and sets all the ports in its VRID VLAN to Blocking.

If a failover occurs, the Backup becomes the new Master and changes all its VRID ports to the Forwarding state.

Other devices can use the redundant paths provided by the VSRP devices. In this example, three devices use the redundant paths. A device that is not itself configured for VSRP but is connected to a device that is configured for VSRP, is ***VSRP aware***. In this example, the three devices connected to the VSRP devices are VSRP aware. A device that is VSRP aware can failover its link to the new Master in sub-second time, by changing the MAC address associated with the redundant path.

When you configure VSRP, make sure each of the non-VSRP devices connected to the VSRP devices has a separate link to each of the VSRP devices.

NOTE
If the PowerConnect is configured as the VSRP Master and it is connected to a FastIron switch (FESX, FSX, SuperX, FGS, and FLS) that is operating as a VSRP-Aware device, the FastIron switch must have the **vsrp-aware tc-vlan-flush** command configured at the VLAN level.
When the **vsrp-aware tc-vlan-flush** command is enabled on the FastIron switch, MAC addresses will be flushed at the VLAN level, instead of at the port level. MAC addresses will be flushed for every topology change (TC) received on the VSRP-aware ports.

Layer 2 redundancy

VSRP provides Layer 2 redundancy. This means that Layer 2 links are backed up.

You can configure VSRP to provide redundancy for Layer 2 only or also for Layer 3:

- Layer 2 only – The Layer 2 links are backup up but specific IP addresses are not backed up.
- Layer 2 and Layer 3 – The Layer 2 links are backup up and a specific IP address is also backed up. Layer 3 VSRP is the same as VRRPE. However, using VSRP provides redundancy at both layers at the same time.

The PowerConnect supports Layer 2 and Layer 3 redundancy. You can configure a PowerConnect for either Layer 2 only or Layer 2 and Layer 3. To configure for Layer 3, specify the IP address you are backing up.

NOTE
If you want to provide Layer 3 redundancy only, disable VSRP and use VRRPE.

Master election and failover

Each VSRP device advertises its VSRP priority in Hello messages. During Master election, the VSRP device with the highest priority for a given VRID becomes the Master for that VRID. After Master election, the Master sends Hello messages at regular intervals to inform the Backups that the Master is healthy.

If there is a tie for highest VSRP priority, the tie is resolved as follows:

- PowerConnect devices – The PowerConnect whose virtual routing interface has a higher IP address becomes the master.

VSRP failover

Each Backup listens for Hello messages from the Master. The Hello messages indicate that the Master is still available. If the Backups stop receiving Hello messages from the Master, the election process occurs again and the Backup with the highest priority becomes the new Master.

Each Backup waits for a specific period of time, the Dead Interval, to receive a new Hello message from the Master. If the Backup does not receive a Hello message from the Master by the time the Dead Interval expires, the Backup sends a Hello message of its own, which includes the Backup's VSRP priority, to advertise the Backup's intent to become the Master. If there are multiple Backups for the VRID, each Backup sends a Hello message.

When a Backup sends a Hello message announcing its intent to become the Master, the Backup also starts a hold-down timer. During the hold-down time, the Backup listens for a Hello message with a higher priority than its own:

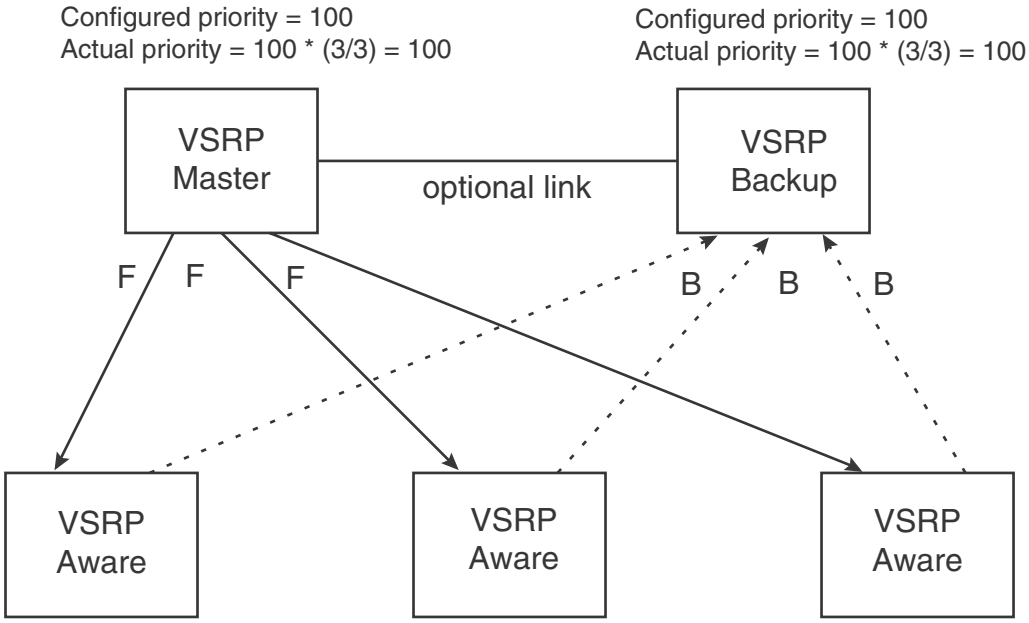
- If the Backup receives a Hello message with a higher priority than its own, the Backup resets its Dead Interval and returns to normal Backup status.
- If the Backup does not receive a Hello message with a higher priority than its own by the time the hold-down timer expires, the Backup becomes the new Master and starts forwarding Layer 2 traffic on all ports.

If you increase the timer scale value, each timer's value is divided by the scale value. To achieve sub-second failover times, you can change the scale to a value up to 10. This shortens all the VSRP timers to 10 percent of their configured values.

VSRP priority calculation

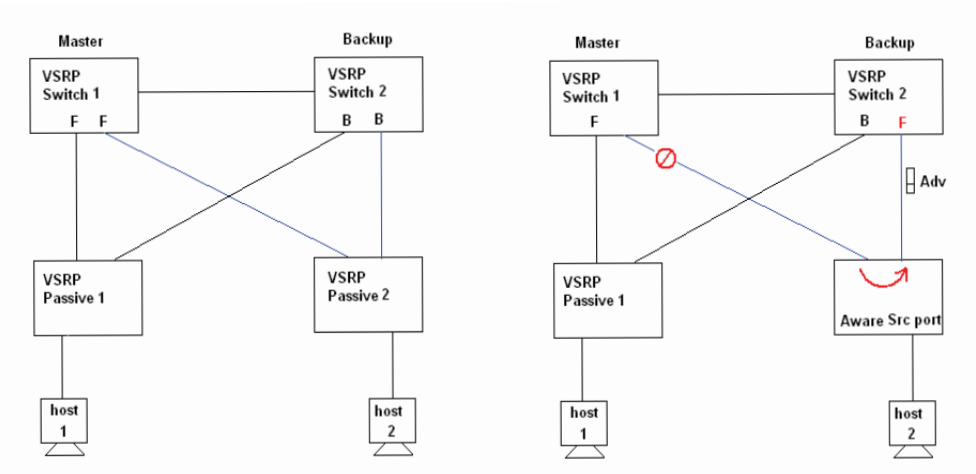
Each VSRP device has a VSRP priority for each VRID and its VLAN. The VRID is used during Master election for the VRID. By default, a device's VSRP priority is the value configured on the device (which is 100 by default). However, to ensure that a Backup with a high number of up ports for a given VRID is elected, the device reduces the priority if a port in the VRID's VLAN goes down. For example, if two Backups each have a configured priority of 100, and have three ports in VRID 1 in VLAN 10, each Backup begins with an equal priority, 100. This is shown in Figure 93

FIGURE 93 VSRP priority



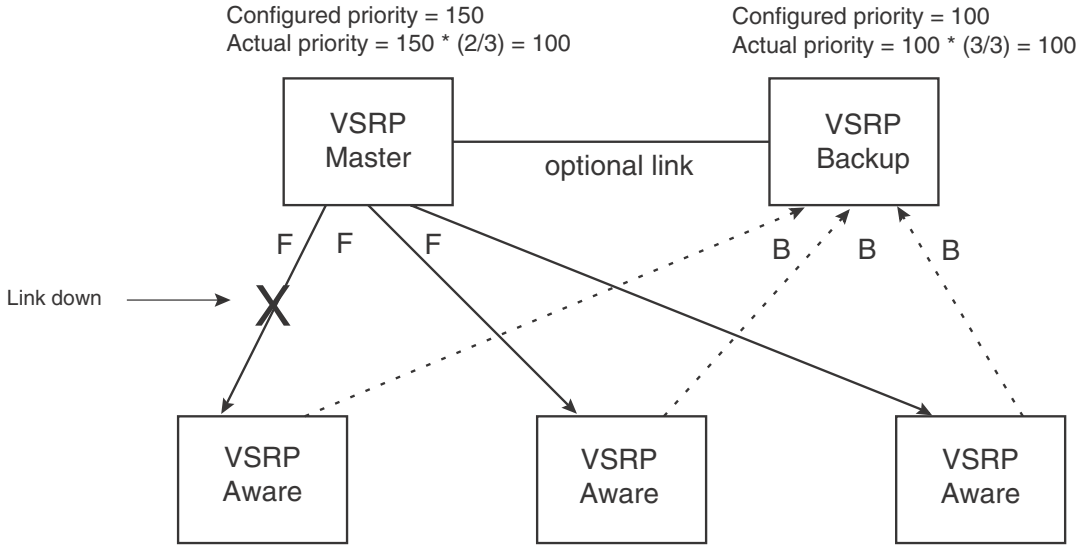
However, if one of the VRID's ports goes down on one of the Backups, that Backup's priority is reduced. If the Master's priority is reduced enough to make the priority lower than a Backup's priority, the VRID fails over to the Backup. Figure 94 shows an example.

FIGURE 94 VSRP priority recalculation



You can reduce the sensitivity of a VSRP device to failover by increasing its configured VSRP priority. For example, you can increase the configured priority of the VSRP device on the left in Figure 94 to 150. In this case, failure of a single link does not cause failover. The link failure caused the priority to be reduced to 100, which is still equal to the priority of the other device. This is shown in Figure 95.

FIGURE 95 VSRP priority bias

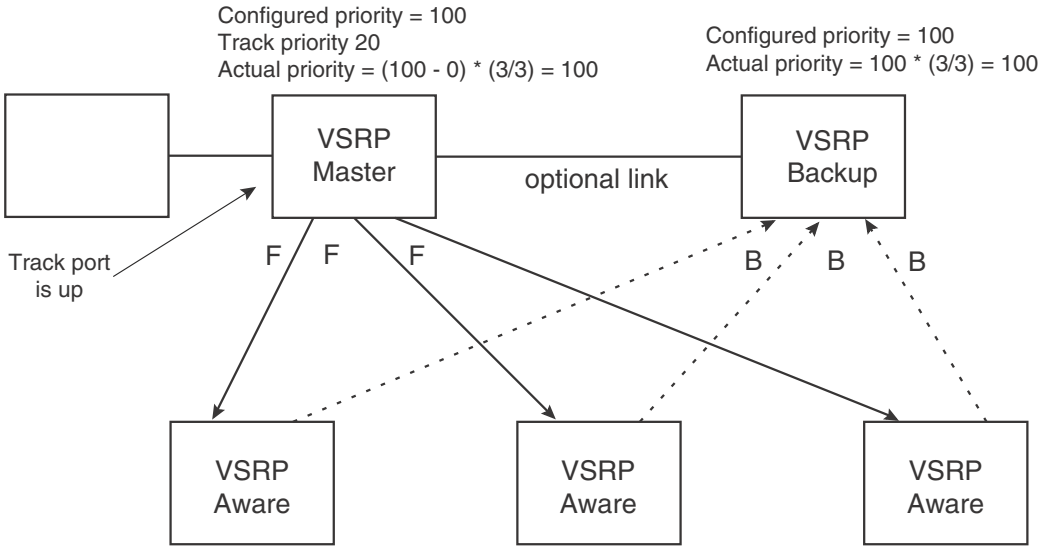


Track ports

Optionally, you can configure track ports to be included during VSRP priority calculation. In VSRP, a ***track port*** is a port that is not a member of the VRID's VLAN, but whose state is nonetheless considered when the priority is calculated. Typically, a track port represents the exit side of traffic received on the VRID ports. By default, no track ports are configured.

When you configure a track port, you assign a priority value to the port. If the port goes down, VSRP subtracts the track port's priority value from the configured VSRP priority. For example, if the you configure a track port with priority 20 and the configured VSRP priority is 100, the software subtracts 20 from 100 if the track port goes down, resulting in a VSRP priority of 80. The new priority value is used when calculating the VSRP priority. Figure 96 shows an example.

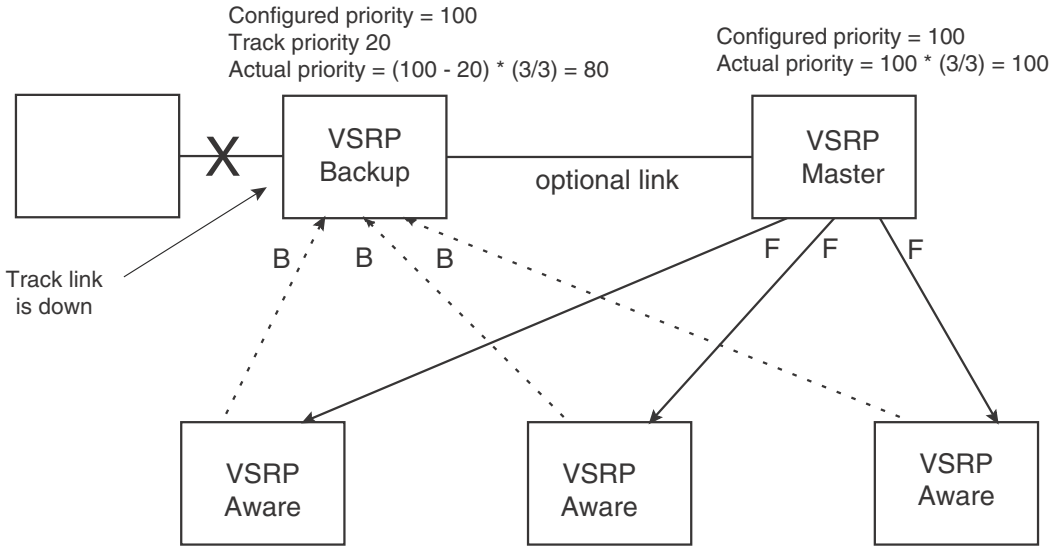
FIGURE 96 Track port priority



In Figure 96, the track port is up. SInce the port is up, the track priority does not affect the VSRP

priority calculation. If the track port goes down, the track priority does affect VSRP priority calculation, as shown in Figure 97.

FIGURE 97 Track port priority subtracted during priority calculation



MAC address failover on VSRP-aware devices

VSRP-aware devices maintain a record of each VRID and its VLAN. When the device has received a Hello message for a VRID in a given VLAN, the device creates a record for that VRID and VLAN and includes the port number in the record. Each subsequent time the device receives a Hello message for the same VRID and VLAN, the device checks the port number:

- If the port number is the same as the port that previously received a Hello message, the VSRP-aware device assumes that the message came from the same VSRP Master that sent the previous message.
- If the port number does not match, the VSRP-aware device assumes that a VSRP failover has occurred to a new Master, and moves the MAC addresses learned on the previous port to the new port.

The VRID records age out if unused. This can occur if the VSRP-aware device becomes disconnected from the Master. The VSRP-aware device will wait for a Hello message for the period of time equal to the following:

VRID Age = Dead Interval + Hold-down Interval + (3 x Hello Interval)

The values for these timers are determined by the VSRP device sending the Hello messages. If the Master uses the default timer values, the age time for VRID records on the VSRP-aware devices is as follows:

3 + 2 + (3 x 1) = 8 seconds

In this case, if the VSRP-aware device does not receive a new Hello message for a VRID in a given VLAN, on any port, the device assumes the connection to the Master is unavailable and removes the VRID record.

Configuring basic VSRP parameters

To configure VSRP, perform the following required tasks:

- Configure a port-based VLAN containing the ports for which you want to provide VSRP service.

  NOTE
  If you already have a port-based VLAN but only want to use VSRP on a sub-set of the VLANs ports, you can selectively remove ports from VSRP service in the VLAN. Refer to "Removing a port from the VRID's VLAN" on page 530.

- Configure a VRID:
  - Specify that the device is a backup. Since VSRP, like VRRPE, does not have an "owner", all VSRP devices are backups. The active device for a VRID is elected based on the VRID priority, which is configurable.
  - Enable VSRP on the VRID.

The following example shows a simple VSRP configuration.

```
NetIron(config)# vlan 200
NetIron(config-vlan-200)# tag ethernet 1/1 to 1/8
NetIron(config-vlan-200)# vsrp vrid 1
NetIron(config-vlan-200-vrid-1)# backup
NetIron(config-vlan-200-vrid-1)# enable
```

Syntax: [no] vsrp vrid <*num*>

The <*num*> parameter specifies the VRID and can be from 1 – 255.

Syntax: [no] backup [priority <*value*>**] [track-priority** <*value*>**]**

This command is required. In VSRP, all devices on which a VRID are configured are Backups. The Master is then elected based on the VSRP priority of each device. There is no "owner" device as there is in VRRP.

For information about the command's optional parameters, refer to the following:

- "Changing the backup priority" on page 530
- "Changing the default track priority" on page 533

Syntax: [no] enable | disable

Note on VSRP support when using ESI

VSRP is supported only for VLANs that are part of the default ESI. VSRP is not supported for VLANs configured under user-defined ESIs.

Configuring optional VSRP parameters

The following sections describe how to configure optional VSRP parameters.

Enabling Layer 3 VSRP

Layer 2 VSRP is enabled globally by default on the device; it just needs to be activated or enabled on a VRID. If you want to use Layer 3 VSRP, you must enable it by entering the following command at the CONFIG level.

```
NetIron(config)# router vsrp
```

Syntax: [no] router vsrp

If you want to provide Layer 3 redundancy only, you could use VRRP or VRRP-Extended. You may use **router vrrp** or **router vrrp-extended** as long as **router vsrp** is not enabled.

Disabling or re-enabling VSRP

To disable Layer 3 VSRP, enter the following command at the global CONFIG level.

```
NetIron(config)# no router vsrp
router vsrp is disabled. All vsrp config data will be lost when writing to flash
```

To re-enable the protocol, enter the following command.

```
NetIron(config)# router vsrp
```

Syntax: [no] router vsrp

Configuring authentication

If the interfaces on which you configure the VRID use authentication, the VSRP packets on those interfaces also must use the same authentication. VSRP supports the following authentication types:

- **No authentication** – The interfaces do not use authentication. This is the default.
- **Simple** – The interfaces use a simple text-string as a password in packets sent on the interface. If the interfaces use simple password authentication, the VRID configured on the interfaces must use the same authentication type and the same password.

To configure a simple password, enter a command such as the following at the interface configuration level.

```
NetIron(config-if-e10000-1/6)# ip vsrp auth-type simple-text-auth ourpword
```

This command configures the simple text password “ourpword”.

Syntax: [no] ip vsrp auth-type no-auth | simple-text-auth <*auth-data*>

The **auth-type no-auth** parameter indicates that the VRID and the interface it is configured on do not use authentication.

The **auth-type simple-text-auth** <*auth-data*> parameter indicates that the VRID and the interface it is configured on use a simple text password for authentication. The <*auth-data*> value is the password. If you use this parameter, make sure all interfaces on all the devices supporting this VRID are configured for simple password authentication and use the same password.

VSRP 2

In VSRP setup, there are always at least two VSRP switches for each VSRP instance. A passive device should always have either one access link or one trunk link connected with each VSRP switch for each VSRP instance. This can create a black hole scenario. A black hole is when VSRP failover causes data traffic from the switches/hosts which connect to VSRP passive switch to go nowhere.

VSRP 2 can detect the health of each pair/set of links for each VSRP instance.VSRP 2 detects which link of VSRP backup switch not receiving any advertisement for specific time duration. The VSRP backup switch can treat the link of the pair on the master side is down or broken and set its own link of the pair in forwarding state. The VSRP backup switch sends out gratuitous ARP for VSRP master only to this link. Other links of other VSRP instances in VSRP backup switch are still in blocking state as shown in figure Figure 98,Figure 99, and Figure 100.

FIGURE 98 Black hole scenario 1

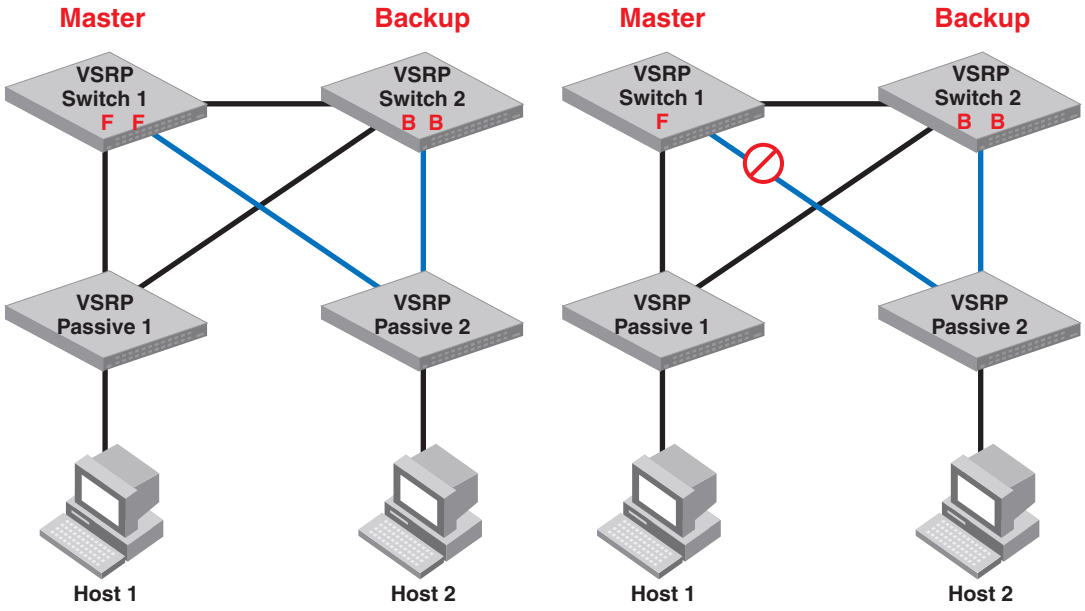


FIGURE 99 Black hole scenario 2

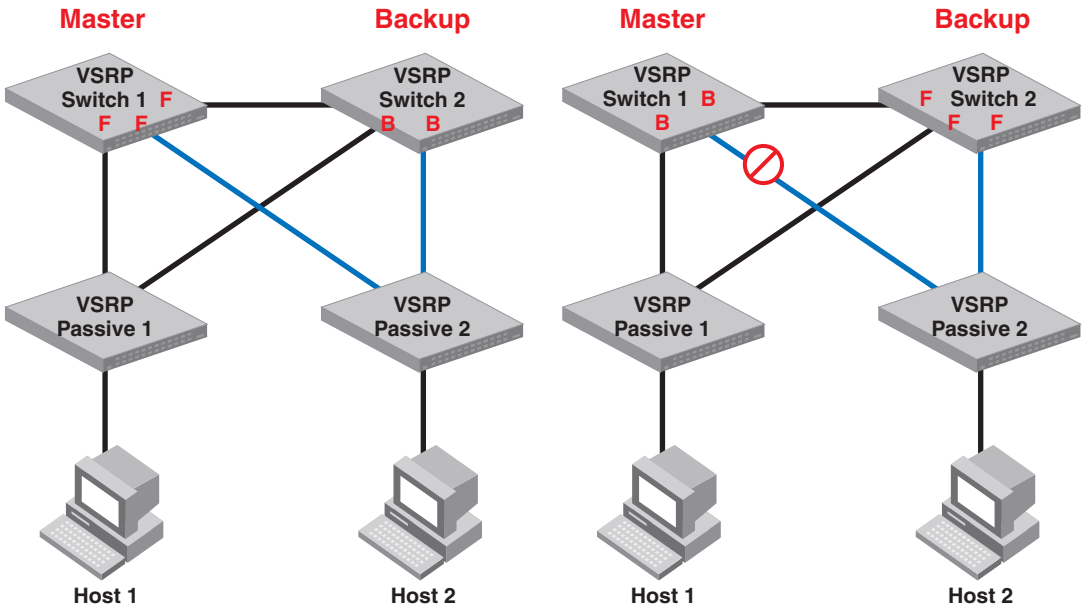
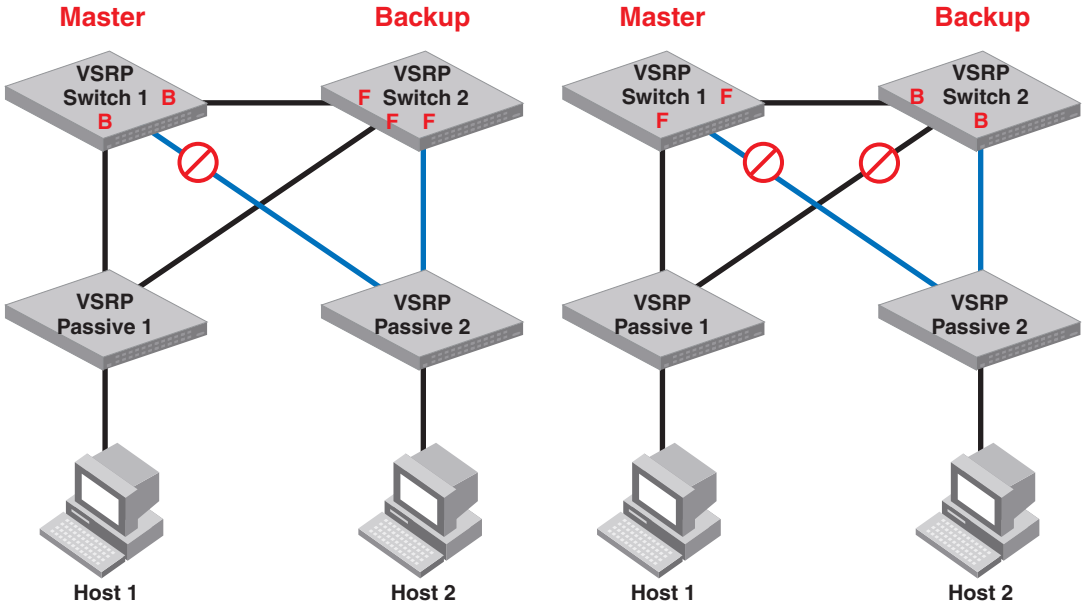


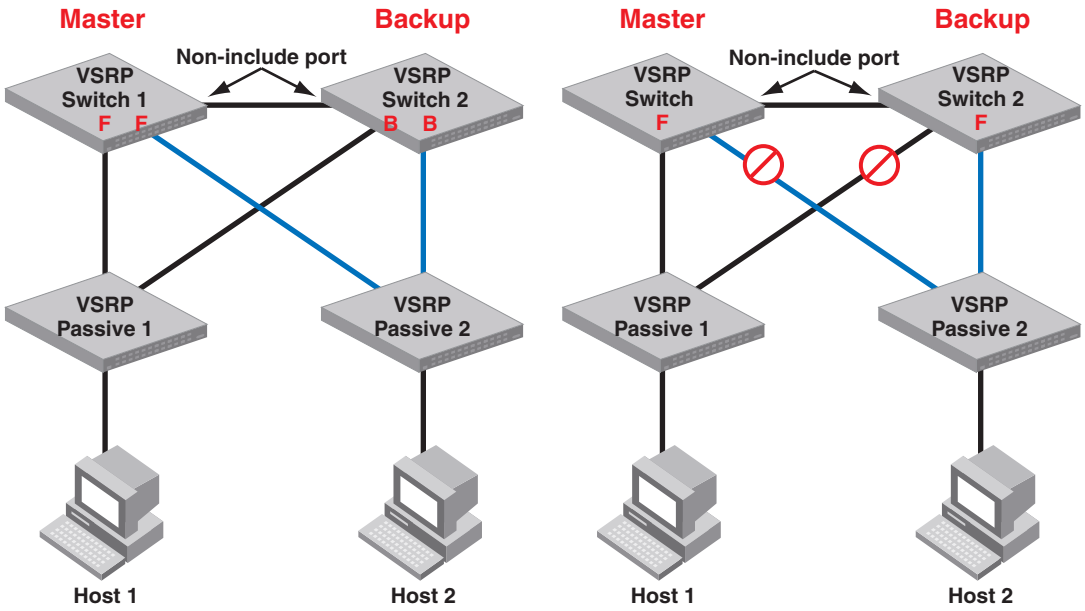
FIGURE 100 Black hole scenario 3



VSRP failove:

- VSRP backup set all include links in blocking state. Blocking ports drop data traffic.
- VSRP failover changes master state by current priority change.
- Current priority changes by link failure and track port failure.

FIGURE 101 Correct VSRP behavior



VSRP is switch redundancy, VSRP 2 is link redundancy.

When VSRP backup changes an include port from blocking state to forwarding state, to make the aware session changes, VSRP backup will send advertisements on the forwarding include ports every 3*hello time.

VSRP aware switches are able to change the src port of aware session and flush MACs.

For VSRP non-aware switches (other vendors), the non-aware switch will flush MACs because the link connecting VSRP master is failed.

VSRP backup will toggle the interface when it sets an include port to forwarding state by VSRP 2.

VSRP 2 doesn't change the master/backup state, only changes the port state.

The change of Master/backup state (VSRP failover) still follows the rules of current priority of VSRP.

VSRP 2 supports:

- hold-down time
- track port
- preempt-mode
- restart port
- topology groups
- VLAN groups.

- VPLS VLAN by topology group.
- L3 VSRP with a condition: non-include link in between two VSRP routers is a must.

Configuration considerations:

- If multiple VSRP instances on multiple VLAN are configured the on one side of link pair, the same VSRP instances of same VLANs must configure on another side of link pair.
- Link-pair has to be enabled or disabled on all VSRP switches for the same VSRP instance.
- Currently, VSRP 2 only supports two VSRP switches in the topology. Multiple VSRP switches may cause a loop when the link redundancy set the VSRP ports in the forwarding state in the link failed cases.
- For VSRP 2 supporting Layer 3 VSRP, it is necessary to have a non-include link in between two VSRP switches. VSRP virtual router is still in VSRP master. Layer 3 data traffic is switched by VSRP backup to VSRP master. The traffic is routed by VSRP master (virtual router).

Configuring VSRP 2

Configure VSRP see"Configuring basic VSRP parameters" on page 522 , then use the **link-redundancy** command at the interface level to enable link redundancy. To enable link redundancy, use commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# vsrp vrid 10
NetIron(config-vlan-10-vsrp-vrid-10)# link-redundancy
```

After enabling link redundancy, VSRP switches to master-confirm state and the backup state creates a link redundant port list.

VSRP switches that are in initial state and master state don't need to create link redundant port list

Syntax: [no] vsrp-linkpair-id <id>

Displaying VSRP 2

To display VSRP, use the **show vsrp** command as shown below.

```
NetIron#show vsrp
VLAN 10
 Auth-type no authentication
 VRID 1
 ========
  State           Administrative-status Advertise-backup Preempt-mode Link-Red
  Backup          Enabled               Enabled          True         Enabled

  Parameter        Configured Current    Unit/Formula
  Priority         100        100        (100-0)*(3.0/3.0)
  Hello-interval   1          1           sec/10
  Dead-interval    3          3           sec/10
  Hold-interval    3          3           sec/10
  Initial-ttl      2          2           hops
  Backup-Hello     600        600         sec/10

  Master router 219.243.150.0 or MAC xxxx.dbf3.9600 expires in 00:00:03
  Member ports:      ethe 2/14 ethe 2/18 to 2/19
```

```
  Operational ports: ethe 2/14 ethe 2/18 to 2/19
   Forwarding ports:  ethe 2/14
Link-Redundancy-port:
  port 2/14 status FORWARD
  port 2/18 status BLOCK
  port 2/19 status BLOCK
```

Syntax: show vsrp [vrid <*num*> **| vlan** <*vlan-id*>**]**

This display shows the following information when you use the **vrid** <*num*> or **vlan** <*vlan-id*> parameter. For information about the display when you use the **aware** parameter, refer to "Displaying the active interfaces for a VRID" on page 537.

TABLE 94 CLI display of VSRP VRID or VLAN information

This field...	Displays...
Total number of VSRP routers defined	The total number of VRIDs configured on this device.
VLAN	The VLAN on which VSRP is configured.
auth-type	The authentication type in effect on the ports in the VSRP VLAN.
VRID parameters	
VRID	The VRID for which the following information is displayed.
state	This device's VSRP state for the VRID. The state can be one of the following: • initialize – VSRP is not enabled on the VRID. If the state remains "initialize" after you enable VSRP on the VRID, make sure that the VRID is also configured on the other routers Routing Switches and that the routers Routing Switches can communicate with each other. **NOTE:** If the state is "initialize" and the mode is incomplete, make sure you have specified the IP address for the VRID. • standby – This device is a Backup for the VRID. • master – This device is the Master for the VRID.
Administrative-status	The administrative status of the VRID. The administrative status can be one of the following: • disabled – The VRID is configured on the interface but VSRP or VRRPE has not been activated on the interface. • enabled – VSRP has been activated on the interface.
Advertise-backup	Whether the device is enabled to send VSRP Hello messages when it is a Backup. This field can have one of the following values: • disabled – The device does not send Hello messages when it is a Backup. • enabled – The device does send Hello messages when it is a Backup.
Preempt-mode	Whether the device can be pre-empted by a device with a higher VSRP priority after this device becomes the Master. This field can have one of the following values: • disabled – The device cannot be pre-empted. • enabled – The device can be pre-empted.

NOTE: For the following fields:

- Configured – indicates the parameter value configured on this device.
- Current – indicates the parameter value received from the Master.
- Unit – indicates the formula used tor calculating the VSRP priority and the timer scales in effect for the VSRP timers. A timer's true value is the value listed in the Configured or Current field divided by the scale value.

TABLE 94 CLI display of VSRP VRID or VLAN information (Continued)

This field...	Displays...
priority	The device's preferability for becoming the Master for the VRID. During negotiation, the Backup with the highest priority becomes the Master. If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.
hello-interval	The number of seconds between Hello messages from the Master to the Backups for a given VRID.
dead-interval	The configured value for the dead interval. The dead interval is the number of seconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID. **NOTE:** If the value is 0, then you have not configured this parameter.
hold-interval	The number of seconds a Backup that intends to become the Master will wait before actually beginning to forward Layer 2 traffic for the VRID. If the Backup receives a Hello message with a higher priority than its own before the hold-down interval expires, the Backup remains in the Backup state and does not become the new Master.
initial-ttl	The number of hops a Hello message can traverse after leaving the device before the Hello message is dropped. A Foundry MRP ring counts as one hop, regardless of the number of nodes in the ring.
next hello sent in	The amount of time until the Master's dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this router itself will become the Master. **NOTE:** This field applies only when this device is a Backup.
master router	The IP address of the master router.
Member ports	The ports in the VRID.
Operational ports	The member ports that are currently up.
Forwarding ports	The member ports that are currently in the Forwarding state. Ports that are forwarding on the Master are listed. Ports on the Standby, which are in the Blocking state, are not listed.
Link-Redundancy-port	The status os the port on which link redundancy has been configured.

Removing a port from the VRID's VLAN

By default, all the ports in the VLAN on which you configure a VRID are interfaces for the VRID. You can remove a port from the VRID while allowing it to remain in the VLAN.

Removing a port is useful in the following cases:

- There is no risk of a loop occurring, such as when the port is attached directly to an end host.
- You plan to use a port in a Foundry MRP ring.

To remove a port from a VRID, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# no include-port ethernet 1/2
```

Syntax: [no] include-port ethernet <*slot*>/<*portnum*>

The **ethernet** <*slot*>/<*portnum*> parameter specifies the port you are removing from the VRID. The port remains in the VLAN but its forwarding state is not controlled by VSRP.

Changing the backup priority

When you enter the backup command to configure the device as a VSRP Backup for the VRID, you also can change the backup priority and the track priority:

- The backup priority is used for election of the Master. The VSRP Backup with the highest priority value for the VRID is elected as the Master for that VRID. The default priority is 100. If two or more Backups are tied with the highest priority, the Backup with the highest IP address becomes the Master for the VRID.
- The track priority is used with the track port feature. Refer to "VSRP priority calculation" on page 518 and "Changing the default track priority" on page 533.

To change the backup priority, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# backup priority 75
```

Syntax: [no] backup [priority <*value*>**] [track-priority** <*value*>**]**

The **priority** <*value*> parameter specifies the VRRP priority for this interface and VRID. You can specify a value from 3 – 254. The default is 100.

For a description of the **track-priority** <*value*> parameter, refer to "Changing the default track priority" on page 533.

Saving the timer values received from the Master

The Hello messages sent by a VRID's master contain the VRID values for the following VSRP timers:

- Hello interval
- Dead interval
- Backup Hello interval
- Hold-down interval

By default, each Backup saves the configured timer values to its startup configuration file when you save the device's configuration.

You can configure a Backup to instead save the current timer values received from the Master when you save the configuration. Saving the current timer values instead of the configured ones helps ensure consistent timer usage for all the VRID's devices.

NOTE
The Backups always use the value of the timer scale received from the Master, regardless of whether the timer values that are saved in the configuration are the values configured on the Backup or the values received from the Master.

To configure a Backup to save the VSRP timer values received from the Master instead of the timer values configured on the Backup, enter the following command.

```
NetIron(config-vlan-200-vrid-1)# save-current-values
```

Syntax: [no] save-current-values

Changing the Time-To-Live (TTL)

A VSRP Hello packet's TTL specifies how many hops the packet can traverse before being dropped. A hop can be a router or a Layer 2 Switch. You can specify from 1 – 255. The default TTL is 2. When a VSRP device (Master or Backup) sends a VSRP HEllo packet, the device subtracts one from the TTL. Thus, if the TTL is 2, the device that originates the Hello packet sends it out with a TTL of 1. Each subsequent device that receives the packet also subtracts one from the packet's TTL. When the packet has a TTL of 1, the receiving device subtracts 1 and then drops the packet because the TTL is zero.

NOTE
A Foundry MRP ring is considered to be a single hop, regardless of the number of nodes in the ring.

To change the TTL for a VRID, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# initial-ttl 5
```

Syntax: [no] initial-ttl <*num*>

The <*num*> parameter specifies the TTL and can be from 1 – 255. The default TTL is 2.

Changing the Hello interval

The Master periodically sends Hello messages to the Backups. To change the Hello interval, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# hello-interval 10
```

Syntax: [no] hello-interval <*num*>

The <*num*> parameter specifies the interval and can be from 1 – 84 units of 100 milliseconds. The default is 1 unit of 100 ms.

NOTE
The default Dead interval is three times the Hello interval plus one-half second. Generally, if you change the Hello interval, you also should change the Dead interval on the Backups.

NOTE
If you change the timer scale, the change affects the actual number of seconds.

Changing the Dead interval

The Dead interval is the number of seconds a Backup waits for a Hello message from the Master before determining that the Master is dead. The default is 3 seconds. This is three times the default Hello interval.

To change the Dead interval, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# dead-interval 30
```

Syntax: [no] dead-interval <*num*>

The <*num*> parameter specifies the interval and can be from 1 – 84 seconds. The default is 3 seconds.

NOTE
If you change the timer scale, the change affects the actual number of seconds.

Changing the Backup Hello state and interval

By default, Backups do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup to send Hello messages to the Master, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# advertise backup
```

Syntax: [no] advertise backup

When a Backup is enabled to send Hello messages, the Backup sends a Hello message to the Master every 60 seconds by default. You can change the interval to be up to 3600 seconds.

To change the Backup Hello interval, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# backup-hello-interval 180
```

Syntax: [no] backup-hello-interval <*num*>

The <*num*> parameter specifies the message interval and can be from 60 – 3600 seconds. The default is 60 seconds.

NOTE
If you change the timer scale, the change affects the actual number of seconds.

Changing the hold-down interval

The hold-down interval prevents Layer 2 loops from occurring during failover, by delaying the new Master from forwarding traffic long enough to ensure that the failed Master is really unavailable.

To change the Hold-down interval, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# hold-down-interval 4
```

Syntax: [no] hold-down-interval <*num*>

The <*num*> parameter specifies the hold-down interval and can be from 1 – 84 seconds. The default is 2 seconds.

NOTE
If you change the timer scale, the change affects the actual number of seconds.

Changing the default track priority

When you configure a VRID to track the link state of other interfaces, if one of the tracked interface goes down, the software changes the VSRP priority of the VRID interface.

The software reduces the VRID priority by the amount of the priority of the tracked interface that went down. For example, if the VSRP interface's priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VSRP interface's priority to 40. If another tracked interface goes down, the software reduces the VRID's priority again, by the amount of the tracked interface's track priority.

The default track priority for all track ports is 1. You can change the default track priority or override the default for an individual track port:

- To change the default track priority, use the **backup track-priority** command, described below.
- To override the default track priority for a specific track port, use the **track-port** command. Refer to "Specifying a track port" on page 533.

To change the track priority, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# backup track-priority 2
```

Syntax: [no] backup [priority <*value*>**] [track-priority** <*value*>**]**

Specifying a track port

You can configure the VRID on one interface to track the link state of another interface on the device. This capability is useful for tracking the state of the exit interface for the path for which the VRID is providing redundancy. Refer to "VSRP priority calculation" on page 518.

To configure a VRID to track an interface, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# track-port e 2/4
```

Syntax: [no] track-port ethernet <*slot*>/<*portnum*> **| ve** <*num*> **[priority** <*num*>**]**

The **priority** <*num*> parameter changes the VSRP priority of the interface. If this interface goes down, the VRID's VSRP priority is reduced by the amount of the track port priority you specify here.

NOTE
The priority <*num*> option changes the priority of the specified interface, overriding the default track port priority. To change the default track port priority, use the **backup track-priority** <*num*> command.

Disabling or re-enabling Backup preemption

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the VRID. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the VRID.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, because the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

To disable preemption on a Backup, enter a command such as the following at the configuration level for the VRID.

```
NetIron(config-vlan-200-vrid-1)# non-preempt-mode
```

Syntax: [no] non-preempt-mode

Displaying VSRP information

You can display the following VSRP information:

- Configuration information and current parameter values for a VRID or VLAN
- The interfaces on a VSRP-aware device that are active for the VRID

Displaying VRID information

To display VSRP information, enter the following command.

```
NetIron# show vsrp vrid 10
    VLAN 100
     Auth-type no authentication
     VRID 100
     ========
      State           Administrative-status Advertise-backup Preempt-mode
      Master          Enabled               Disabled         True

      Parameter        Configured Current    Unit/Formula
      Priority         100        100        (100-0)*(3.0/3.0)
      Hello-interval   1          1          sec/10
      Dead-interval    3          3          sec/10
      Hold-interval    3          3          sec/10
      Initial-ttl      2          2          hops
      Next hello sent in 00:00:00
      Member ports:      ethe 1/1 ethe 2/1 ethe 2/10
      Operational ports: ethe 1/1 ethe 2/1 ethe 2/10
```

On a devices where the VSRP Fast Start feature is enabled.

```
NetIron(config-vlan-100-vrid-100)#show vsrp vrid 100
VLAN 100
  auth-type no authentication
  VRID 100
  ========
   State      Administrative-status Advertise-backup Preempt-mode
   master     enabled               disabled         true
   Parameter        Configured Current    Unit/Formula
   priority         100        50         (100-0)*(2.0/4.0)
   hello-interval   1          1          sec/1
   dead-interval    3          3          sec/1
   hold-interval    3          3          sec/1
   initial-ttl      2          2          hops
   next hello sent in 00:00:00.3
   Member ports:      ethe 2/5 to 2/8
   Operational ports: ethe 2/5 ethe 2/8
   Forwarding ports:  ethe 2/5 ethe 2/8
   Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

Syntax: show vsrp [vrid <*num*> **| vlan** <*vlan-id*>**]**

This display shows the following information when you use the **vrid** <*num*> or **vlan** <*vlan-id*> parameter. For information about the display when you use the **aware** parameter, refer to "Displaying the active interfaces for a VRID" on page 537.

TABLE 95 CLI display of VSRP VRID or VLAN information

This field...	Displays...
Total number of VSRP routers defined	The total number of VRIDs configured on this device.
VLAN	The VLAN on which VSRP is configured.
auth-type	The authentication type in effect on the ports in the VSRP VLAN.
VRID parameters	
VRID	The VRID for which the following information is displayed.
state	This device's VSRP state for the VRID. The state can be one of the following: • initialize – VSRP is not enabled on the VRID. If the state remains "initialize" after you enable VSRP on the VRID, make sure that the VRID is also configured on the other routers Routing Switches and that the routers Routing Switches can communicate with each other. **NOTE:** If the state is "initialize" and the mode is incomplete, make sure you have specified the IP address for the VRID. • standby – This device is a Backup for the VRID. • master – This device is the Master for the VRID.
Administrative-status	The administrative status of the VRID. The administrative status can be one of the following: • disabled – The VRID is configured on the interface but VSRP or VRRPE has not been activated on the interface. • enabled – VSRP has been activated on the interface.
Advertise-backup	Whether the device is enabled to send VSRP Hello messages when it is a Backup. This field can have one of the following values: • disabled – The device does not send Hello messages when it is a Backup. • enabled – The device does send Hello messages when it is a Backup.

TABLE 95 CLI display of VSRP VRID or VLAN information (Continued)

This field...	Displays...
Preempt-mode	Whether the device can be pre-empted by a device with a higher VSRP priority after this device becomes the Master. This field can have one of the following values: • disabled – The device cannot be pre-empted. • enabled – The device can be pre-empted.
NOTE: For the following fields: • Configured – indicates the parameter value configured on this device. • Current – indicates the parameter value received from the Master. • Unit – indicates the formula used tor calculating the VSRP priority and the timer scales in effect for the VSRP timers. A timer's true value is the value listed in the Configured or Current field divided by the scale value.	
priority	The device's preferability for becoming the Master for the VRID. During negotiation, the Backup with the highest priority becomes the Master. If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.
hello-interval	The number of seconds between Hello messages from the Master to the Backups for a given VRID.
dead-interval	The configured value for the dead interval. The dead interval is the number of seconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID. **NOTE:** If the value is 0, then you have not configured this parameter.
hold-interval	The number of seconds a Backup that intends to become the Master will wait before actually beginning to forward Layer 2 traffic for the VRID. If the Backup receives a Hello message with a higher priority than its own before the hold-down interval expires, the Backup remains in the Backup state and does not become the new Master.
initial-ttl	The number of hops a Hello message can traverse after leaving the device before the Hello message is dropped. **NOTE:** A Foundry MRP ring counts as one hop, regardless of the number of nodes in the ring.
next hello sent in	The amount of time until the Master's dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this router itself will become the Master. **NOTE:** This field applies only when this device is a Backup.
master router	The IP address of the master router.
Member ports	The ports in the VRID.
Operational ports	The member ports that are currently up.
Forwarding ports	The member ports that are currently in the Forwarding state. Ports that are forwarding on the Master are listed. Ports on the Standby, which are in the Blocking state, are not listed.

Displaying the active interfaces for a VRID

On a VSRP-aware device, you can display VLAN and port information for the connections to the VSRP devices (Master and Backups).

To display the active VRID interfaces, enter the following command on the VSRP-aware device.

```
NetIron(config-vlan-200-vrid-1)# show vsrp aware

Aware port listing
VLAN ID  VRID  Last Port
100      1     3/2
200      2     4/1
```

Syntax: show vsrp aware

This display shows the following information when you use the **aware** parameter. For information about the display when you use the **vrid** <*num*> or **vlan** <*vlan-id*> parameter, refer to "Displaying VRID information" on page 534.

TABLE 96 CLI display of VSRP-aware information

This field...	Displays...
VLAN ID	The VLAN that contains the VSRP-aware device's connection with the VSRP Master and Backups.
VRID	The VRID.
Last Port	The most recent active port connection to the VRID. This is the port connected to the current Master. If a failover occurs, the VSRP-aware device changes the port to the port connected to the new Master. The VSRP-aware device uses this port to send and receive data through the backed up node.

VSRP fast start

It allows non-Dell or non-VSRP aware devices that are connected to a Dell device that is the VSRP Master to quickly switch over to the new Master when a VSRP failover occurs

This feature causes the port on a VSRP Master to restart when a VSRP failover occurs. When the port shuts down at the start of the restart, ports on the non-VSRP aware devices that are connected to the VSRP Master flush the MAC address they have learned for the VSRP master. After a specified time, the port on the previous VSRP Master (which now becomes the Backup) returns back online. Ports on the non-VSRP aware devices switch over to the new Master and learn its MAC address.

Special considerations when configuring VSRP fast start

Consider the following when configuring VSRP fats start:

- VSRP is sensitive to port status. When a port goes down, the VSRP instance lowers its priority based on the port up fraction. (refer to "VSRP priority calculation" on page 518 for more information on how priority is changed by port status). Since the VSRP fast start feature toggles port status by bringing ports down and up it can affect VSRP instances because their priorities get reduced when a port goes down. To avoid this, the VSRP fast start implementation keeps track of ports that it brings down and suppresses port down events for these ports (as concerns VSRP).

- Once a VSRP restart port is brought up by a VSRP instance, other VSRP instances (in Master state) that have this port as a member do not go to forwarding immediately. This is a safety measure that is required to prevent transitory loops. This could happen if a peer VSRP node gets completely cut off from this node and assumed Master state. In this case, where there are 2 VSRP instances that are in Master state and forwarding, the port comes up and starts forwarding immediately. This would cause a forwarding loop. To avoid this, the VSRP instance delays forwarding.

Recommendations for configuring VSRP fast start

The following recommendations apply to configurations where multiple VSRP instances are running between peer devices sharing the same set of ports:

- Multiple VSRP instances configured on the same ports can cause VSRP instances to be completely cut off from peer VSRP instances. This can cause VSRP instances to toggle back and forth between master and backup mode. For this reason, we recommend that you configure VSRP fast start on a per port basis rather than for the entire VLAN.
- We recommend that VSRP peers have a directly connected port without VSRP fast start enabled on it. This allows protocol control packets to be received and sent even if other ports between the master and standby are down.
- The VSRP restart time should be configured based on the type of connecting device since some devices can take a long time to bring a port up or down (as long as several seconds). In order to ensure that the port restart is registered by neighboring device, the restart time may need to be changed to a value higher than the default value of 1 second.

Configuring VSRP fast start

The VSRP fast start feature can be enabled on a VSRP-configured device, either on the VLAN to which the VRID of the VSRP-configured device belongs (globally) or on a port that belongs to the VRID.

To globally configure a VSRP-configured device to shut down its ports when a failover occurs, then restart after five seconds, enter the following command.

```
NetIron(configure)# vlan 100
NetIron(configure-vlan-100)# vsrp vrid 1
NetIron(configure-vlan-100-vrid-1)# restart-ports 5
```

Syntax: [no] restart-ports <*seconds*>

This command shuts down all the ports that belong to the VLAN when a failover occurs. All the ports will have the specified VRID.

To configure a single port on a VSRP-configured device to shut down when a failover occurs, then restart after a period of time, enter the following command.

```
NetIron(configure)# interface ethernet 1/1
NetIron(configure-if-1/1)# vsrp restart-port 5
```

Syntax: [no] vsrp restart-port <*seconds*>

In both commands, the <*seconds*> parameter instructs the VSRP Master to shut down its port for the specified number of seconds before it starts back up. Enter a value between 1 – 120 seconds. The default is 1 second.

Displaying ports that have VSRP fast start feature enabled

The **show vsrp vrid** command shows the ports on which the VSRP fast start feature is enabled.

```
NetIron(config-vlan-100-vrid-100)#show vsrp vrid 100
VLAN 100
  auth-type no authentication
  VRID 100
  ========
   State      Administrative-status Advertise-backup Preempt-mode save-current
   master     enabled               disabled         true         false
   Parameter        Configured Current    Unit/Formula
   priority         100        50         (100-0)*(2.0/4.0)
   hello-interval   1          1          sec/1
   dead-interval    3          3          sec/1
   hold-interval    3          3          sec/1
   initial-ttl      2          2          hops
   next hello sent in 00:00:00.3
   Member ports:      ethe 2/5 to 2/8
   Operational ports: ethe 2/5 ethe 2/8
   Forwarding ports:  ethe 2/5 ethe 2/8
   Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

The "Restart ports:" line lists the ports that have the VSRP fast start enabled, and the downtime for each port. Refer to Table 95 on page 535 to interpret the remaining information on the display.

VSRP slow start

In a VSRP configuration, if a Master router goes down, the Backup router with the highest priority takes over. When the Master comes back up again, it takes over from the Backup. By default, this transition from Backup back to Master takes place immediately. You can configure the VSRP slow start timer feature, which causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup. (This range is currently set to between 1 to 600 ticks (1/10 second to 60 seconds). This interval allows time for VSRP convergence when the Master is restored.

To set the VSRP slow start timer to 30 seconds, enter the following command.

```
NetIron(config)# router vsrp
NetIron(config-vsrp-router)# slow-start 300
```

Syntax: [no] slow-start <ticks>

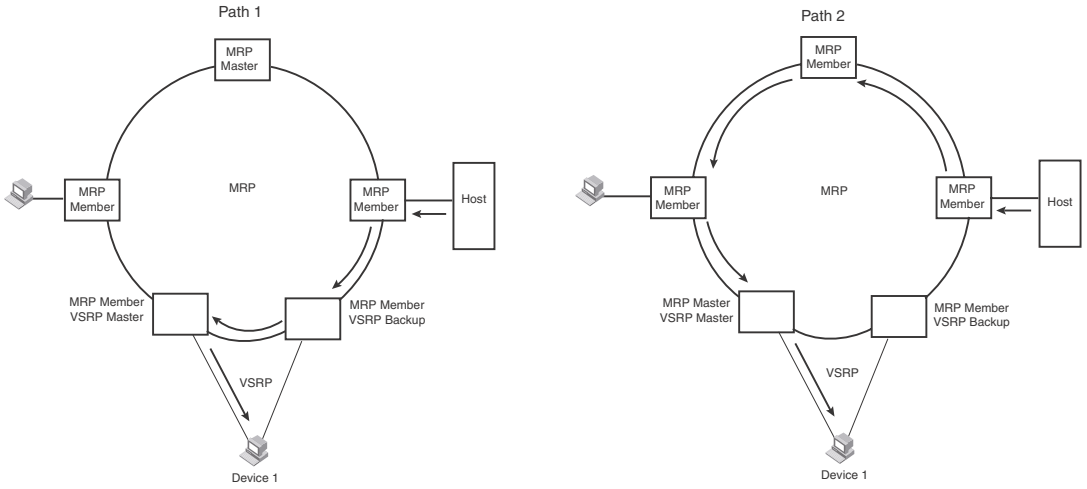
The ticks parameter can range is from 1 to 600 ticks (1/10 second to 60 seconds).

When the VSRP slow start timer is enabled, if the Master goes down, the Backup takes over immediately. If the Master subsequently comes back up again, the amount of time specified by the VSRP slow start timer elapses (in this example, 30 seconds) before the Master takes over from the Backup.

VSRP and Foundry MRP signaling

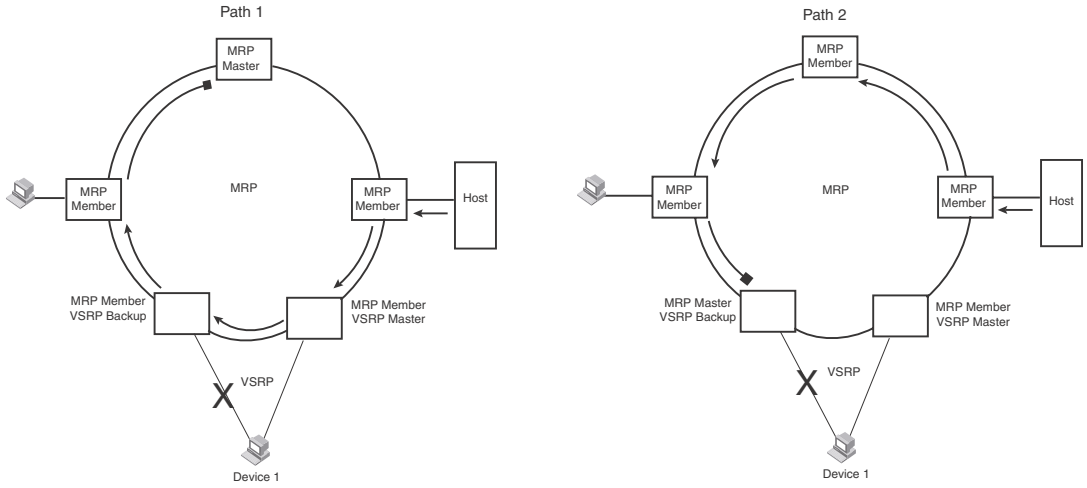
A device may connect to a Foundry MRP ring through VSRP to provide a redundant path between the device and the Foundry MRP ring. VSRP and Foundry MRP signaling, ensures rapid failover by flushing MAC addresses appropriately. The host on the Foundry MRP ring learns the MAC addresses of all devices on the Foundry MRP ring and VSRP link. From these MAC addresses, the host creates a MAC database (table), which is used to establish a data path from the host to a VSRP-linked device. Figure 102 below shows two possible data paths from the host to Device 1.

FIGURE 102 Two data paths from host on a Foundry MRP ring to a VSRP-linked device



If a VSRP failover from master to backup occurs, VSRP needs to inform Foundry MRP of the topology change; otherwise, data from the host continues along the obsolete learned path and never reach the VSRP-linked device, as shown in Figure 103.

FIGURE 103 VSRP on Foundry MRP rings that failed over

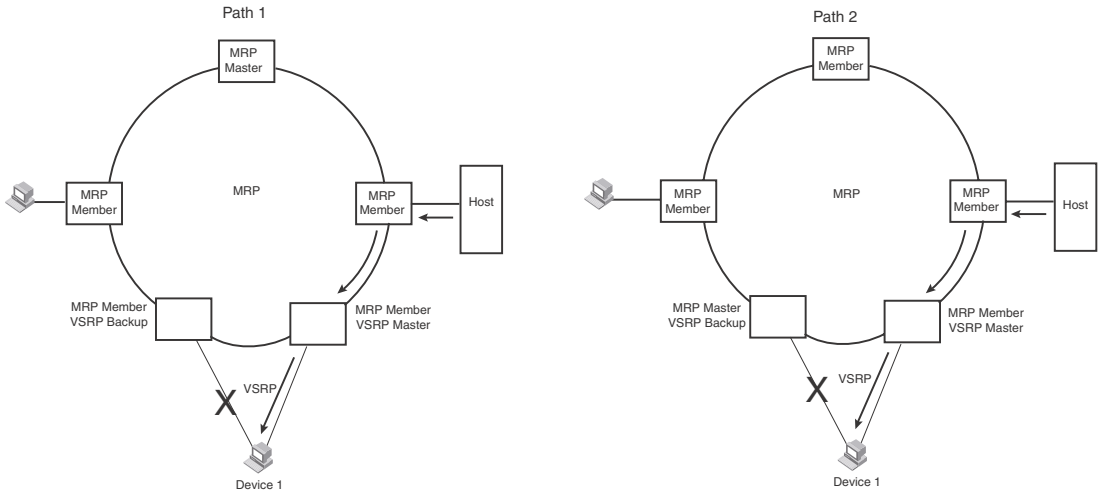


To ensure that Foundry MRP is informed of the topology change and to achieve convergence rapidly, a signaling process for the interaction between VSRP and Foundry MRP. When a VSRP node fails, a new VSRP master is selected. The new VSRP master finds all Foundry MRP instances impacted by the failover. Then each Foundry MRP instance does the following:

- The Foundry MRP node sends out a Foundry MRP PDU with the mac-flush flag set three times on the Foundry MRP ring.
- The Foundry MRP node that receives this Foundry MRP PDU empties all the MAC address entries from its interfaces that participate on the Foundry MRP ring.
- The Foundry MRP node then forwards the Foundry MRP PDU with the mac-flush flag set to the next Foundry MRP node that is in forwarding state.

The process continues until the Master Foundry MRP node's secondary (blocking) interface blocks the packet. Once the MAC address entries have been flushed, the MAC table can be rebuilt for the new path from the host to the VSRP-linked device (Figure 104).

FIGURE 104 New path established



There are no CLI commands used to configure this process.

Topology Groups 16

Overview

The following Topology Group features are supported by the NetIron MLX Series devices.

- Topology Groups
- Master VLAN and Member VLANs
- Master VLANs and Customer VLANs in Foundry MRP
- Control Ports and Free Ports
- Dual tag support for VPLS VLANs
- Adding VPLS VLANs to Topology Groups

A topology group is a named set of VLANs that share a Layer 2 control protocol. Topology groups simplify configuration and enhance scalability of Layer 2 protocols by allowing you to run a single instance of a Layer 2 protocol on multiple VLANs. One instance of the Layer 2 protocol controls all the VLANs.

For example, if a PowerConnect is deployed in a Metro network and provides forwarding for two Foundry MRP rings that each contain 128 VLANs, you can configure a topology group for each ring. If a link failure in a ring causes a topology change, the change is applied to all the VLANs in the ring's topology group. Without topology groups, you would need to configure a separate ring for each VLAN.

You can use topology groups with the following Layer 2 protocols:

- STP
- Foundry MRP
- VSRP
- RSTP

Master VLAN and member VLANs

Each topology group contains a master VLAN and can contain one or more member VLANs and VLAN groups. A definition for each of these VLAN types follows:

- **Master VLAN** – The master VLAN contains the configuration information for the Layer 2 protocol. For example, if you plan to use the topology group for Foundry MRP, the topology group's master VLAN contains the ring configuration information.
- **Member VLANs** – The member VLANs are additional VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the member VLANs. A change to the master VLAN's Layer 2 protocol configuration or Layer 2 topology affects all the member VLANs. Member VLANs do not independently run a Layer 2 protocol. VPLS VLANs can become member VLANs within a topology group.

- **Member VLAN groups** – A VLAN group is a named set of VLANs. The VLANs within a VLAN group have the same ports and use the same values for other VLAN parameters.

When a Layer 2 topology change occurs on a port in the master VLAN, the same change is applied to that port in all the member VLANs that contain the port. For example, if you configure a topology group whose master VLAN contains ports 1/1 and 1/2, a Layer 2 state change on port 1/1 applies to port 1/1 in all the member VLANs that contain that port. However, the state change does not affect port 1/1 in VLANs that are not members of the topology group.

Master VLANs and customer VLANs in Foundry MRP

A topology group enables you to control forwarding in multiple VLANs using a single instance of a Layer 2 protocol such as Foundry MRP. For more information on topology group and Foundry MRP, refer to "Master VLANs and member VLANs in a topology group" on page 461.

Control ports and free ports

A port in a topology group can be a control port or a free port:

- **Control port** – is a port in the master VLAN and, therefore, is controlled by the Layer 2 protocol configured in the master VLAN. The same port in all the member VLANs is controlled by the master VLAN's Layer 2 protocol. Each member VLAN must contain all of the control ports. All other ports in the member VLAN are "free ports."
- **Free port** – is not controlled by the master VLAN's Layer 2 protocol. The master VLAN can contain free ports. (In this case, the Layer 2 protocol is disabled on those ports.) In addition, any ports in the member VLANs that are not also in the master VLAN are free ports.

NOTE
Because free ports are not controlled by the master port's Layer 2 protocol, they are assumed always to be in the forwarding state, when enabled.

Configuration considerations

The configuration considerations are as follows:

- You can configure up to 255 topology groups. Each group can control up to 4095 VLANs. A VLAN cannot be controlled by more than one topology group.
- The topology group must contain a master VLAN and can also contain individual member VLANs, VLAN groups, or a combination of individual member VLANs and VLAN groups. Therefore, you configure the master VLAN and member VLANs or member VLAN groups before you configure a topology group.
- After you add a VLAN as a member of a topology group, the router deletes all the Layer 2 protocol information on that VLAN.
- If you add a new master VLAN to a topology group that already has a master VLAN, the new master VLAN replaces the older master VLAN. All member VLANs and VLAN groups follow the Layer 2 protocol settings of the new master VLAN.

- If you remove the master VLAN (by entering **no master-vlan** <*vlan-id*>), the software selects the new master VLAN from member VLANs. A new candidate master VLAN will be in configured order to a member VLAN so that the first added member VLAN will be a new candidate master VLAN. Once you save and reload, a member VLAN with the youngest VLAN ID will be a new candidate master. The new master VLAN inherits the Layer 2 protocol settings of the older master VLAN.
- After you add a VLAN or VLAN group as a member of a topology group, all the Layer 2 protocol configuration information for the VLAN or group is deleted. For example, if STP is configured on a VLAN and you add the VLAN to a topology group, the STP configuration is removed from the VLAN. After you add the VLAN to a topology group, the VLAN uses the Layer 2 protocol settings of the master VLAN.

  If you remove a member VLAN or VLAN group from a topology group, you need to reconfigure the Layer 2 protocol information in the VLAN or VLAN group.
- On platforms where the Ethernet Service Instance (ESI) framework is supported, master VLANs in a topology group must either be in the default ESI or within the same ESI. Master and member VLANs cannot span multiple ESIs.

Configuring a topology group

To configure a topology group, enter commands such as the following.

```
NetIron(config)# topology-group 2
NetIron(config-topo-group-2)# master-vlan 2
NetIron(config-topo-group-2)# member-vlan 3
NetIron(config-topo-group-2)# member-vlan 4
NetIron(config-topo-group-2)# member-vlan 5
NetIron(config-topo-group-2)# member-group 2
```

The commands configure topology group 2 and add the following to it:

- VLAN 2 as master VLAN
- VLANs 3, 4, and 5 as member VLANs
- Member VLAN group 2

Syntax: [no] topology-group <*group-id*>

The command creates a topology group.The <*group-id*> parameter assigns an ID 1 – 255 to the topology group.

Syntax: [no] master-vlan <*vlan-id*>

This command adds the master VLAN to the topology group. The VLAN must already be configured. Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

NOTE
When a port is added to a master VLAN, it will be added as a free port. Similarly when a port has to be removed from master VLAN, first disable any the Layer 2 protocol on the port, then remove the port from the master VLAN.

Syntax: [no] member-vlan <*vlan-id*>

This command adds a member VLAN to the topology group. The VLAN must already be configured.

Syntax: [no] member-group <*num*>

This command adds a VLAN group to the topology group. The <*num*> specifies a VLAN group ID. The VLAN group must already be configured.

Adding VPLS VLANs to topology groups

To add *single-tagged or untagged* VPLS VLANs as member VLANs to a topology group, use the **member-vlan vpls** command as shown in the following example.

```
NetIron(config)# topology-group 2
NetIron(config-topo-group-2)# master-vlan 2
NetIron(config-topo-group-2)# member-vlan vpls id 34 vlan 42 to 45
```

To add *dual-tagged* VPLS VLANs as member VLANs to a topology group, use the **member-vlan vpls** command as shown in the following configuration example.

```
NetIron(config)# topology-group 1
NetIron(config-topo-group-1)# master-vlan 10
NetIron(config-topo-group-1)# master-vlan 20
NetIron(config-topo-group-1)# member-vlan vpls id 5 vlan 300 inner-vlan 20 to 25
```

Syntax: [no] member-vlan vpls [id <*vpls-id*> **| name** <*vpls-name*> **}** **vlan** <*vlan-id*> **[to** <*vlan-id*> **]**

OR

Syntax: [no] member-vlan vpls [**id** <*vpls-id*> | **name** <*vpls-name*>] **vlan** <*vlan-id*> [**inner-vlan** <*inner-vlan-id*> [**to** <*inner-vlan-id*>]] ....

The **id** option allows you to specify the VPLS instance that you are configuring into the topology group by using the VPLS ID of the instance. A value in the range of 1 - 4294967294 can be entered for VPLS ID.

The **name** option allows you to specify the VPLS instance that you are configuring into the topology group by using the name of the instance.

The <*vlan-id*> variable is used with the **vlan** keyword to specify the VPLS VLAN being configured into topology group. You can specify multiple <*vlan-id*> values or specify a range of VLANs using the **to** option.

The **inner-vlan** option allows you to specify a VPLS dual-tagged (double-tagged) VLAN configuration.

NOTE
The **inner-vlan** option does not allow both outer VLAN ranges and inner VLAN ranges for a given VPLS instance. Once an outer VLAN range is specified, the inner VLAN option is not allowed. However, if a single outer VLAN is specified, the inner VLAN option and range is allowed.

NOTE
You cannot delete a topology master VLAN if the topology group has only VPLS VLAN members and no L2 VLAN members because the normal procedure for deleting a topology master VLAN is to elect another L2 VLAN as the new master. Because a VPLS VLAN cannot be a master VLAN, you must have at least one L2 VLAN as a member. If it does not currently exist, you must add a L2 VLAN before deleting a topology master.

Topology group support within an ESI

Topology groups can be configured with VLANs that are part of a user-defined ESI. (Consult Table • on page 543 to see which platform supports topology groups within an ESI.) When you configure topology groups in such a scenario, both the master and member VLANs must be part of the same ESI. If an ESI is not specified, the system assumes a reference to the default ESI. Below is an example of configuring topology groups with VLANs that are part of a user-defined ESI.

```
NetIron(config)# topology-group 2
NetIron(config-topo-group-2)# master-vlan service-esi 2
NetIron(config-topo-group-2)# member-vlan service-esi 3
NetIron(config-topo-group-2)# member-vlan service-esi 4
NetIron(config-topo-group-2)# member-vlan service-esi 5
NetIron(config-topo-group-2)# member-group service-esi 2
```

The commands configure topology group 2 and add the following to it:

- VLAN 2 in ESI "service-esi" as master VLAN
- VLANs 3, 4, and 5 in ESI "service-ESI" as member VLANs
- Member VLAN group 2

Syntax: [no] topology-group <*group-id*>

This command creates a topology group. The <*group-id*> parameter assigns an ID in the range 1 – 255 to the topology group.

Syntax: [no] master-vlan esi-name <*vlan-id*>

This command adds the master VLAN in ESI identified by the VLAN ID to the topology group. The VLAN must already be configured in the ESI "esi-name". Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

Syntax: [no] member-vlan esi-name <*vlan-id*>

This command adds a member VLAN in ESI identified by the VLAN ID to the topology group. The VLAN must already be configured in the ESI "esi-name".

Syntax: [no] member-group esi-name <*num*>

This command adds a VLAN group in ESI identified by “esi-name” to the topology group. The <*num*> specifies a VLAN group ID. The VLAN group must already be configured.

Displaying topology group information

This section contains examples of the **show topology-group** output. Support for topology groups within an ESI is supported on a minority of platforms (listed in Table on page 547), so its example appears at the end of this section.

Displaying topology group information on a NetIron MLX series router

The **show topology-group** command offers a choice between one of two mandatory parameters. The command syntax is as follows.

Syntax: show topology-group <*group-id*> | **hw-index-table** [<*hw-index*>]

The first example in this section utilizes the first possible mandatory parameter, <*group-id*>. The second example utilizes the second possible mandatory parameter, **hw-index-table**, along with an optional variable, a hardware index number.

Display topology group information by using a Group ID

To display topology group information for group 10, enter the **show topology-group** command.

```
NetIron#show topology-group 10
Topology Group 10
=================
Topo HW Index   : 0
Master VLAN     : 10
VPLS VLAN exist : TRUE
Member VLAN     : 20
Member Group  : None
Member VPLSs  : vpls id 200 vlan 200 200 to 250
                vpls name vpls100 vlan 300 inner-vlan 5
                vpls id 5 vlan 300
                vpls id 5 vlan 300 inner-vlan 5
                vpls id 5 vlan 300 inner-vlan 20 to
Control Ports : ethe 3/11 to 3/12 ethe 3/15 to 3/16
Free Ports :
```

Syntax: show topology-group <*group-id*>

This display shows the following information:

TABLE 97 CLI display of topology group information

This field...	Displays...
Topology Group	The ID of the topology group. The range for <*group-id*> is 1 – 256.
Topo HW Index	A topology hardware index is a unique hardware ID that is assigned to a VLAN when an L2 protocol is configured on the VLAN. The VLAN that runs the L2 protocol could be a standalone L2 VLAN or a master VLAN under a topology group. The range for <*hw-index*> is 0 – 511. (The **show topology-group hw-index-table** output shows the mapping of a topology hardware index to a VLAN.)
Master-VLAN	The master VLAN for the topology group. The settings for STP, Foundry MRP, RSTP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
VPLS VLAN exist	Indicates whether the topology group has one or more VPLS VLANs as a topology group member. The content of this field is TRUE or FALSE.
Member-VLAN	The VLAN ID of the member of the topology group.
Member VPLSs	The VPLS VLAN members in the topology group.
Control ports	The master VLAN ports that are configured with Layer 2 protocol information. The Layer 2 protocol configuration and state of these ports in the master VLAN applies to the same port numbers in all the member VLANs.
Free ports	A list of all free ports in the topology group. A free port is not controlled by the Layer 2 protocol information in the master VLAN. In the example screen output, the absence of any number indicates that no ports are free.

Display topology group information by using hardware index table numbers

Display the information for hardware index table 0..

```
NetIron#show topology-group hw-index-table  0
Total Instances : 512
Free Instances  : 511
Topo HW Index    Vlan ID
-------------    -------
0                10
```

Syntax: show topology-group hw-index-table [<*hw-index*>]

The range for <*hw-index*> is 0 – 511. If you do not specify a number for <*hw-index*>, the output screen lists all entries.

TABLE 98 Topology group information with hardware index table

This field...	Displays...
Total Instances	Total number of topology hardware indexes that have been initialized in the system.
Free Instances	Number of free topology hardware indexes that are left in the system.
Topology HW Index	A topology hardware index is assigned to a VLAN when an L2 protocol is configured on the VLAN. The VLAN that runs the L2 protocol could be a standalone L2 VLAN or a master VLAN under a topology group. The **show topology-group hw-index-table** output shows the mapping of a topology hardware index to a VLAN. The range for is 0 – 511. In the example, hardware index table 0 is mapped to the VLAN with an ID of 10.
VLAN ID	The ID of the port-based VLAN that owns the protocol instance on that VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on a device, all protocol information is for VLAN 1.

Configuring VRRP and VRRP-E 17

Overview

The following VRRP and VRRP-E features are supported by the Brocade NetIron XMR Series.

- Standard VRRP
- VRRP Extended (VRRP-E)
- VRRP v2 Authentication
- VRRPv3 for IPv4 and IPv6
- VRRP-E v6
- VRRP alongside RIP
- VRRP alongside OSPF
- VRRP alongside BGP4
- VRRP Track Port
- VRRP Track Priority
- VRRP Backup Preempt
- VRRP Master Router Abdication and Reinstatement
- VRRP-Extended Slow Start
- VRRP-Extended Scale Timer
- VRRP-E Extension for Server Virtualization
- Virtual MAC address per VRID

This chapter describes how to configure the following router redundancy protocols:

- ***Virtual Router Redundancy Protocol (VRRP)*** – The standard router redundancy protocol described in RFC 3768. The devices support VRRP version 2 and VRRP version 3. VRRP v2 supports IPv4 environment and VRRP v3 supports IPv4 and IPv6 environment.
- ***VRRP Extended (VRRP-E)*** – A proprietary version of VRRP that overcomes limitations in the standard protocol. This protocol works only with Dell IP devices. The devices support VRRP-E version 2 and VRRP-E version 3. VRRP-E v2 supports IPv4 environment and VRRP-E v3 supports IPv6 environment.

NOTE
The maximum number of VRRP instances (vrid) supported per interface is 12 for both VRRP and VRRP-E.

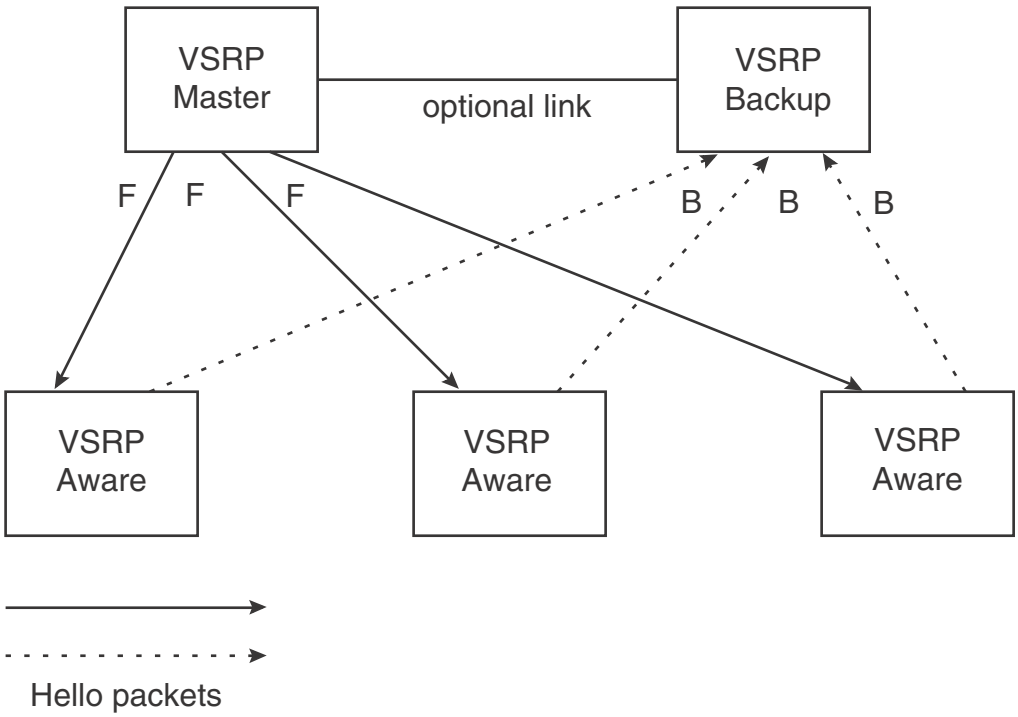
Overview of VRRP

This section presents the standard VRRP options and the options that were added in its implementation of VRRP.

Standard VRRP

VRRP is an election protocol that provides redundancy to routers within a LAN. VRRP allows you to provide alternate router paths for a host without changing the IP address or MAC address by which the host knows its gateway. Consider the situation shown in Figure 105.

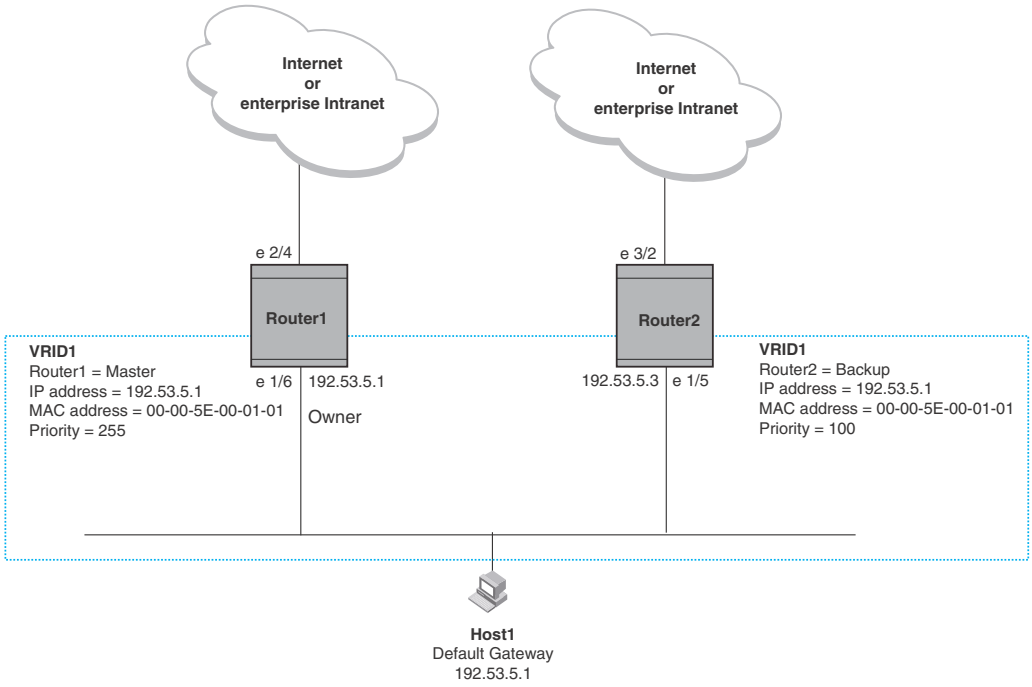
FIGURE 105 Router1 is Host1's default gateway but is a single point of failure



As shown in this example, Host1 uses 192.53.5.1 on Router1 as the host's default gateway out of the subnet. If this interface goes down, Host1 is cut off from the rest of the network. Router1 is thus a single point of failure for Host1's access to other networks.

If Router1 fails, you could configure Host1 to use Router2. Configuring one host with a different default gateway might not require too much extra administration. However, consider a more realistic network with dozens or even hundreds of hosts per subnet; reconfiguring the default gateways for all the hosts is impractical. It is much simpler to configure a VRRP virtual router on Router1 and Router2 to provide a redundant path for the hosts. If VRRP is enabled as in Figure 106, Router 2 provides the default gateway out of the subnet if Router 1 fails.

FIGURE 106 Router1 and Router2 configured as a VRRP virtual routers forredundant network access for Host1



With VRRP, you configure virtual routers that span across the physical routers. A virtual router acts as a default router for hosts on a shared LAN. For example, Figure 106 has one virtual router configured identified as VRID1.This virtual router ID is associated with Router 1 and Router 2.

Since there are more than one IP addresses configured on Router 1 and Router 2, one of the physical addresses is assigned to the virtual router. For example, in Figure 106, IP address 192.53.5.1, the IP address assigned to Router 1's interface 1/6, is assigned as the IP address of virtual router VRID1. Router 1 becomes the Owner of the virtual router VRID1 and is the router that responds to packets addresses to any of the IP addresses in virtual router VRID1.

In addition, one router in the virtual router is elected as the Master router. Other routers act as backups. The Master router is the one that forwards packets sent to the IP addresses in the virtual router and answers ARP requests for these IP addresses. The Backup router takes over for the Master router when the Master router fails.

NOTE
You can provide more redundancy by also configuring a second VRID with Router2 as the Owner and Router1 as the Backup. This type of configuration is sometimes called Multigroup VRRP.

Master router election

Virtual routers use the VRRP priority values associated with each VRRP router to determine which router becomes the Master. When you configure an Owner router, the PowerConnect automatically sets the its VRRP priority to 255, the highest VRRP priority. The router in the virtual router with the highest priority becomes the Master. Other routers become the backup and can be assigned priorities from 3 through 254. The default priority value is 100.

Virtual routers use VRID Hello messages to determine if a Master router is available. They send Hello messages to IP Multicast address 224.0.0.18 at a specified frequency. The Backup routers waits for a duration of time for a Hello message from the Master. This duration is called the Dead Interval. If a Backup router does not receive a Hello message by the time the dead interval expires, the Backup router assumes that the Master router is dead. The Backup router with the highest priority becomes the Master router. Once the Owner router becomes available, it becomes the Master router and the current Master router returns to being a backup router.

Pre-emption

If the pre-emption feature is enabled, a Backup router that is acting as the Master can be pre-empted by another Backup router that has a higher priority. This can occur the if you add a new Backup while the Owner is still available and new Backup router has a higher priority than the Backup router that is acting as Master.

Virtual router MAC address

When you configure a VRID, the software automatically assigns its MAC address as the virtual router's MAC address. The first five octets of the address are the standard MAC prefix for VRRP packets, as described in RFC 3768. The last octet is the VRID. THE VRID number becomes the final octet in the virtual router's virtual MAC address. For example, the MAC address for VRID is 000.5e00.0101.

When the virtual router becomes the Master router, it broadcasts a gratuitous ARP request containing the virtual router's MAC address for each IP address associated with the virtual router. In Figure 106, Router1 sends a gratuitous ARP with MAC address 00-00-5e-00-01-01 and IP address 192.53.5.1. Hosts use the virtual router's MAC address in routed traffic they send to their default IP gateway (in this example, 192.53.5.1).

Enhancements to VRRP

Dell has enhanced VRRP by adding the following options:

- Configuring unique virtual MAC addresses per VRID
- Track Ports and Track Priority
- Suppression of RIP Advertisements for Backed Up Interfaces
- Authentication
- VRRP operation is independent of RIP, OSPF, and BGP

Configuring unique virtual MAC addresses per VRID

In addition to system-configured standards-based virtual MAC addresses, you can manually configure a unique virtual MAC address for each IPv4 and IPv6 VRRP instance, per VRID. For NetIron MLX platform, you can configure a maximum of 2000 virtual MAC addresses.

If there is no manually configured virtual MAC address for a VRRP instance, the system automatically assigns one based on the VRRP RFC standard.

This feature is subject to the following limitations:

- This feature does not support configurable VRRP virtual MAC addresses over MCT.
- This feature has no impact on short-path forwarding for VRRP-E.

NOTE
System-assigned virtual MAC addresses and manually configured virtual MAC addresses can exist at the same time on the device under the same VRID, however the configured value takes precedence. When the configured value is deleted, the assigned value again applies.

To configure a unique VRRP or VRRP-E virtual MAC address for a VRID, complete the following steps.

1. To configure an IPv4 virtual MAC address for VRID 1 (for example), enter the following command at the configure vrid level of the CLI:

   ```
   NetIron(config-if-e1000-1/110-vrid-1)virtual-mac aaaa.bbbb.cccc
   ```

   Syntax: [**no**] **virtual-mac** <*mac-address*>

   Use the **no** version of this command to remove the configured address.

2. To configure an IPv6 virtual MAC address for VRID 1 (for example), enter the following command at the configure vrid level of the CLI:

   ```
   NetIron(config-if-e1000-1/110-ipv6-vrid-1)virtual-mac aaaa.bbbb.cccc
   ```

   Syntax: [**no**] **virtual-mac** <*ipv6 mac-address*>

   Use the **no** version of this command to remove the configured address.

3. To display IPv4 VRRP virtual MAC address configuration information about VRID 1 (for example), enter the following command:

   ```
   NetIron#show ip vrrp vrid 1

   Interface 1/1
   ----------------
   auth-type no authentication

   VRID 1 (index 1)
    interface 1/1
    state master
    administrative-status enabled
    version v2
    mode owner
    virtual mac aaaa.bbbb.cccc (configured)
    priority 255
    current priority 255
    track-priority 2
    hello-interval 1 sec
    backup hello-interval 60 sec
   ```

```
 ip-address 10.20.1.100
```

4. To display IPv4 VRRP-E virtual MAC address configuration information about VRID 1 (for example), enter the following command:

```
NetIron#show ip vrrp-extended vrid 1

Interface 1/1
---------------
auth-type no authentication

VRID 1 (index 1)
 interface 1/1
 state master
 administrative-status disabled
 mode non-owner(backup)
 virtual mac aaaa.bbbb.cccc (configured)
 priority 100
 current priority 100
 track-priority 5
 hello-interval 1 sec
 backup hello-interval 60 sec
 advertise backup disabled
 dead-interval 0 ms
 preempt-mode true
 virtual ip address 10.20.1.100
 short-path-forwarding disabled
```

5. To display IPv6 VRRP virtual MAC address configuration information for VRID 1 (for example), enter the following command:

```
NetIron#show ipv6 vrrp vrid 1

Interface 1/1
---------------
auth-type no authentication

VRID 1 (index 1)
 interface 1/1
 state master
 administrative-status enabled
 version v3
 mode non-owner(backup)
 virtual mac dddd.eeee.ffff (configured)
 priority 100
 current priority 100
 track-priority 1
 hello-interval 1000 ms
 backup hello-interval 60000 ms
 advertise backup disabled
 dead-interval 3600 ms
 preempt-mode true
 ipv6 address 10:20:1::100
 next hello sent in 400 ms
```

6. To display IPv6 VRRP-E virtual MAC address configuration information for VRID 1 (for example), enter the following command:

```
NetIron#show ipv6 vrrp-extended vrid 1

Interface 1/1
```

```
---------------
auth-type no authentication

VRID 1 (index 1)
 interface 1/1
 state master
 administrative-status enabled
 mode non-owner(backup)
 virtual mac dddd.eeee.ffff (configured)
 priority 100
 current priority 100
 track-priority 5
 hello-interval 1 sec
 backup hello-interval 60 sec
 advertise backup disabled
 dead-interval 0 ms
 preempt-mode true
 virtual ipv6 address 10:20:1::100
```

You can also identify configured virtual MAC addresses by entering the **show running-config** command, as shown in this example.

```
NetIron# show running-config interface ethernet 1/11
interface ethernet 1/11
enable
ip ospf area 0
ip address 15.1.1.15/24
```

Syntax: show running-config interface <*slot/port*>

Track ports and track priority

Dell enhanced VRRP by giving a VRRP router the capability to monitor the state of the interfaces on the other end of the route path through the router. For example, in Figure 106 on page 553, interface e1/6 on Router1 owns the IP address to which Host1 directs route traffic on its default gateway. The exit path for this traffic is through Router1's e2/4 interface.

Suppose interface e2/4 goes down. Even if interface e1/6 is still up, Host1 is cut off from other networks. In conventional VRRP, Router1 would continue to be the Master router despite the unavailability of the exit interface for the path the router is supporting. However, if you configure interface e1/6 to track the state of interface e2/4, if e2/4 goes down, interface e1/6 responds by changing Router1's VRRP priority to the value of the track priority. In the configuration shown in Figure 106 on page 553, Router1's priority changes from 255 to 20. One of the parameters contained in the Hello messages the Master router sends to its Backups is the Master router's priority. If the track port feature results in a change in the Master router's priority, the Backup routers quickly become aware of the change and initiate a negotiation for Master router.

In Figure 106 on page 553, the track priority results in Router1's VRRP priority becoming lower than Router2's VRRP priority. As a result, when Router2 learns that it now has a higher priority than Router1, Router2 initiates negotiation for Master router and becomes the new Master router, thus providing an open path for Host1's traffic. To take advantage of the track port feature, make sure the track priorities are always lower than the VRRP priorities. The default track priority for the router that owns the VRID IP addresses is 2. The default track priority for Backup routers is 1. If you change the track port priorities, make sure you assign a higher track priority to the Owner of the IP addresses than the track priority you assign on the Backup routers.

Suppression of RIP advertisements for backed up interfaces

The implementation also enhances VRRP by allowing you to configure the protocol to suppress RIP advertisements for the backed up paths from Backup routers. Normally, a VRRP Backup router includes route information for the interface it is backing up in RIP advertisements. As a result, other routers receive multiple paths for the interface and might sometimes unsuccessfully use the path to the Backup rather than the path to the Master. If you enable the implementation of VRRP to suppress the VRRP Backup routers from advertising the backed up interface in RIP, other routers learn only the path to the Master router for the backed up interface.

Authentication

For backward compatibility with RFC 2338,'s implementation of VRRP can use simple passwords to authenticate VRRP packets. The VRRP authentication type is not a parameter specific to the VRID. Instead, VRRP uses the authentication type associated with the interfaces on which you define the VRID. For example, if you configure your router interfaces to use a simple password to authenticate traffic, VRRP uses the same simple password and VRRP packets that do not contain the password are dropped. If your interfaces do not use authentication, neither does VRRP.

NOTE
The MD5 authentication type is not supported by VRRP or VRRP-E.

NOTE
Authentication is not supported by VRRP v3.

Forcing a master router to abdicate to a standby router

You can force a VRRP Master to abdicate (give away control) of a virtual router to a Backup by temporarily changing the Master's priority to a value less than the Backup's. When you change a VRRP Owner's priority, the change takes effect only for the current power cycle. The change is not saved to the startup configuration file when you save the configuration and is not retained across a reload or reboot. Following a reload or reboot, the VRRP Owner again has priority 255.

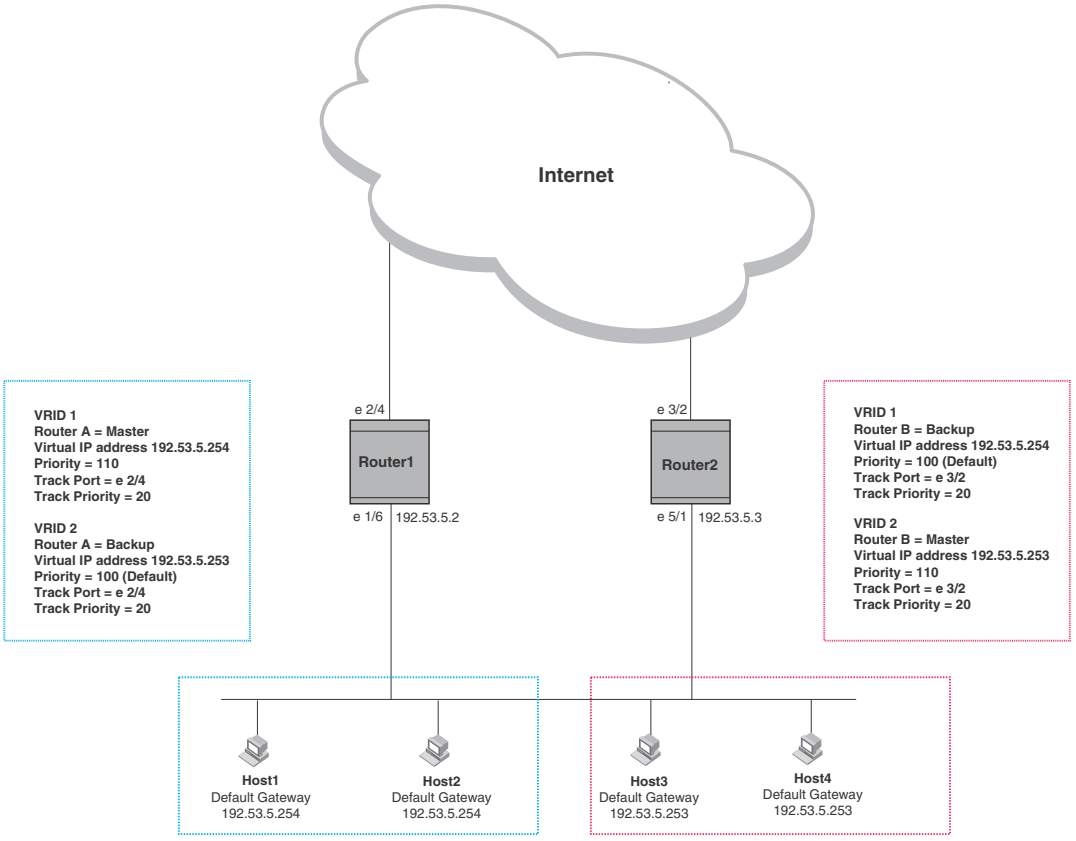
VRRP alongside RIP, OSPF, and BGP4

VRRP operation is independent of the RIP, OSPF, and BGP4 protocols. Their operation is unaffected when VRRP is enabled on a RIP, OSPF, or BGP4 interface.

Overview of VRRP-E

VRRP-E is proprietary version of VRRP that overcomes limitations in the standard protocol. Figure 107 shows an example of a VRRP-E configuration.

FIGURE 107 Router1 and Router2 are configured to provide dual redundant network access for the host



In this example, Router1 and Router2 use VRRP-E to load share as well as provide redundancy to the hosts. The load sharing is accomplished by creating two VRRP-E groups. Each group has its own virtual IP addresses. Half of the clients point to VRID 1's virtual IP address as their default gateway and the other half point to VRID 2's virtual IP address as their default gateway. This will enable some of the outbound Internet traffic to go through Router1 and the rest to go through Router2.

Router1 is the master for VRID 1 (backup priority = 110) and Router2 is the backup for VRID 1 (backup priority = 100). Router1 and Router2 both track the uplinks to the Internet. If an uplink failure occurs on Router1, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the Internet is sent through Router2 instead.

Similarly, Router2 is the master for VRID 2 (backup priority = 110) and Router1 is the backup for VRID 2 (backup priority = 100). Router1 and Router2 are both tracking the uplinks to the Internet. If an uplink failure occurs on Router2, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the internet is sent through Router1 instead.

The PowerConnect configured for VRRP-E can interoperate only with other Dell routers or switches.

ARP behavior with VRRP-E

In the VRRP-E implementation, the source MAC address of the gratuitous ARP sent by the VRRP-E master router will be the VRRP-E virtual MAC address. When the router (either master or backup router) sends an ARP request or reply packet, the sender's MAC address will be the MAC address of the interface on the router. When an ARP request packet for the virtual router IP address is received by the backup router, it will be forwarded to the master router to resolve the ARP. Only master router will answer the ARP request for the virtual router IP address.

Comparison of VRRP and VRRP-E

VRRP-E is similar to VRRP, but differs in the following respects:

- Owners and Backups:
  - VRRP has an Owner and one or more Backups for each virtual router. The Owner is the router that has the IP address used for the virtual router. All the other routers supporting the virtual router are Backups.
  - VRRP-E does not use Owners. All routers are Backups for a given virtual router. The router with the highest priority becomes the Master. If there is a tie for highest priority, the router with the highest IP address becomes the Master. The elected Master owns the virtual IP address and answers ping and ARP requests and so on.
- Master and Backups:
  - VRRP – The “Owner” of the IP address of the VRID is the default Master and has the highest priority (255). The precedence of the Backups is determined by their priorities. The default Master is always the Owner of the IP address of the VRID.
  - VRRP-E – The Master and Backups are selected based on their priority. You can configure any of the PowerConnect devices to be the Master by giving it the highest priority. There is no Owner.
- Virtual Router’s IP address:
  - VRRP requires that the virtual router has an IP address that is configured on the Owner router.
  - VRRP-E requires only that the virtual router’s IP address be in the same subnet as an interface configured on the VRID's interface. In fact, VRRP-E does not allow you to specify an IP address configured on the interface as the VRID IP address.
- VRID's MAC Address:
  - VRRP uses the interfaces’s actual MAC address as the source MAC address. The virtual MAC address for IPv4 VRRP is 00-00-5E-00-01-<*vrid*> and for IPv6 VRRP is 00-00-5E-00-02-<*vrid*>.The <*vrid*> is the ID of the virtual router. The Master owns the Virtual MAC address.
  - VRRP-E uses the interface’s actual MAC address as the source MAC address. The virtual MAC address for IPv4 VRRP-E and IPv6 VRRP-E is 02-E0-52-<*hash-value*>-<*vrid*>, where <*hash-value*> is a two-octet hashed value for the IP address and <*vrid*> is the virtual router ID.

    NOTE
    You cannot reuse the same VRID across IPv4 VRRP-E and IPv6 VRRP-E, if they are in the same broadcast domain.

- Hello packets:
  - VRRP sends Hello messages to IP Multicast address 224.0.0.18.
  - VRRP-E uses UDP to send Hello messages in IP multicast messages. The Hello packets use the interface’s actual MAC address and IP address as the source addresses. The destination MAC address is 01-00-5E-00-00-02, and the destination IP address is 224.0.0.2 (the well-known IP multicast address for “all routers”). Both the source and destination UDP port number is 8888. VRRP messages are encapsulated in the data portion of the packet.

- **Track ports and track priority:**
  - VRRP changes the priority of the VRID to the track priority, which typically is lower than the VRID priority and lower than the VRID's priorities configured on the Backups. For example, if the VRRP interface's priority is 100 and a tracked interface with track priority 20 goes down, the software changes the VRRP interface's priority to 20.
  - VRRP-E reduces the priority of a VRRP-E interface by the amount of a tracked interface's priority if the tracked interface's link goes down. For example, if the VRRP-E interface's priority is 200 and a tracked interface with track priority 20 goes down, the software changes the VRRP-E interface's priority to 180. If another tracked interface goes down, the software reduces the VRID's priority again, by the amount of the tracked interface's track priority.

The most important difference is that all VRRP-E routers are Backups. There is no Owner router. VRRP-E overcomes the limitations in standard VRRP by removing the Owner.

VRRP and VRRP-E parameters

Table 99 lists the VRRP and VRRP-E parameters. Most of the parameters and default values are the same for both protocols. The exceptions are noted in the table.

TABLE 99 VRRP and VRRP-E parameters

Parameter	Description	Default	Refer page...
Protocol	The Virtual Router Redundancy Protocol (VRRP) based on RFC 2338 or VRRP-Extended, Dell's enhanced implementation of VRRP	Disabled **NOTE:** Only one of the protocols can be enabled at a time.	page 564 page 567
VRRP or VRRP-E router	The PowerConnect's active participation as a VRRP or VRRP-E router. Enabling the protocol does not activate the PowerConnect for VRRP or VRRP-E. You must activate the PowerConnect as a VRRP or VRRP-E router after you configure the VRRP or VRRP-E parameters.	Inactive	page 564 page 567
Virtual Router ID (VRID)	The ID of the virtual router you are creating by configuring multiple routers to back up an IP interface. You must configure the same VRID on each router that you want to use to back up the address. No default.	None	page 564 page 567
Virtual Router IP address	This is the address you are backing up. No default. • VRRP – The virtual router IP address must be a real IP address configured on the VRID interface on one of the VRRP routers. This router is the IP address Owner and is the default Master. • VRRP-E – The virtual router IP address must be in the same subnet as a real IP address configured on the VRRP-E interface, but cannot be the same as a real IP address configured on the interface.	None	page 564 page 567

TABLE 99 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	Refer page...
VRID MAC address	The source MAC address in VRRP or VRRP-E packets sent from the VRID interface, and the destination for packets sent to the VRID. • VRRP – A virtual MAC address defined as 00-00-5e-00-01-<*vrid*> for IPv4 VRRP and 00-00-5E-00-02-<vrid> for IPv6 VRRP. The Master owns the Virtual MAC address. • VRRP-E – A virtual MAC address defined as 02-E0-52-<*hash-value*>-<*vrid*> for IPv4 VRRP-E and IPv6 VRRP-E, where <*hash-value*> is a two-octet hashed value for the IP address and <*vrid*> is the ID of the virtual router.	Not configurable	page 554
Authentication type	The type of authentication the VRRP or VRRP-E routers use to validate VRRP or VRRP-E packets. The authentication type must match the authentication type the VRID's port uses with other routing protocols such as OSPF: • No authentication – The interfaces do not use authentication. This is the VRRP default. • Simple – The interface uses a simple text-string as a password in packets sent on the interface. If the interface uses simple password authentication, the VRID configured on the interface must use the same authentication type and the same password. **NOTE:** MD5 is not supported by VRRP or VRRP-E. Authentication is not supported by VRRP v3.	No authentication	page 558 page 569
Router type	Whether the router is an Owner or a Backup: • Owner (VRRP only) – The router on which the real IP address used by the VRID is configured. • Backup – Routers that can provide routing services for the VRID but do not have a real IP address matching the VRID.	VRRP – The Owner is always the router that has the real IP address used by the VRID. All other routers for the VRID are Backups. VRRP-E – All routers for the VRID are Backups.	page 564 page 567
Backup priority	A numeric value that determines a Backup's preferability for becoming the Master for the VRID. During negotiation, the router with the highest priority becomes the Master: • VRRP – The Owner has the highest priority (255); other routers can have a priority from 3 through 254. • VRRP-E – All routers are Backups and have the same priority by default. If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.	VRRP – 255 for the Owner; 100 for each Backup VRRP-E – 100 for all Backups	page 564 page 567
Suppression of RIP advertisements	A router that is running RIP normally advertises routes to a backed up VRID even when the router is not currently the active router for the VRID. Suppression of these advertisements helps ensure that other routers do not receive invalid route paths for the VRID.	Disabled	page 570

TABLE 99 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	Refer page...
Hello interval	The number of seconds or milliseconds between Hello messages from the Master to the Backups for a given VRID. The interval can be from 1 through 84 seconds for VRRP v2 and VRRP-E v2. The interval can be from 100 milliseconds through 8400 milliseconds for VRRP v3 and VRRP-E v3.	One second (VRRP v2 and VRRP-E v2) 1000 milliseconds (VRRP v3 and VRRP-E v3)	page 570
Dead interval	The number of seconds or milliseconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID.	Three times the Hello Interval plus one-half second	page 571
Backup Hello interval	The number of seconds between Hello messages from a Backup to the Master. The message interval can be from 60 through 3600 seconds. You must enable the Backup to send the messages. The messages are disabled by default on Backups. The current Master (whether the VRRP Owner or a Backup) sends Hello messages by default.	Disabled 60 seconds when enabled	page 571
Track port	Another PowerConnect port or virtual interface whose link status is tracked by the VRID's interface. If the link for a tracked interface goes down, the VRRP or VRRP-E priority of the VRID interface is changed, causing the devices to renegotiate for Master.	None	page 557 page 572
Track priority	A VRRP or VRRP-E priority value assigned to the tracked ports. If a tracked port's link goes down, the VRID port's VRRP or VRRP-E priority changes: • VRRP – The priority changes to the value of the tracked port's priority. • VRRP-E – The VRID port's priority is reduced by the amount of the tracked port's priority.	VRRP – 2 VRRP-E – 5	page 557 page 572
Backup preempt mode	Prevents a Backup with a higher VRRP priority from taking control of the VRID from another Backup that has a lower priority but has already assumed control of the VRID.	Enabled	page 573
Slow Start	Causes a specified amount of time to elapse between the time the original Master router is restored and when it takes over from the Backup router. For VRRP-E only.	Disabled	page 574

TABLE 99 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	Refer page...
Scale Timer	Allows you to increase timing sensitivity across all configured or default VRRP-Extended timers. For VRRP-E only.	Disabled	page 574
short-path-forwarding	Enables VRRP-E Extension for Server Virtualization. If enabled, the traffic that is destined to the clients will travel through the short-path forwarding path (dashed line) to reach the client (as shown in Figure 24.4 on page 24-30). Any packets coming from the local subnet of the virtual IP address will be routed to the VRRP-E master router. For example, with VRRP-E Extension for Server Virtualization enabled, the traceroute output will show one extra hop that display the Backup router's interface IP address	Disabled	page 591

Configuring parameters specific to VRRP

VRRP is configured at the interface level. To implement a simple VRRP configuration using all the default values, enter the following commands.

NOTE
When you use the command **router vrrp** to enter the VRRP configuration mode, the command prompt does not change and results in the general configuration command prompt as shown in the following: NetIron`(config)#`. This differs from entering the VRRP extended mode where entering the router **vrrp-extended** command results in a command prompt such as the following: PowerConnect`(config-vrrpe-router)#`

Configuring the VRRP version

You can specify the version for the VRRP instance. For example, use the following command to configure the instance of VRRP to use VRRP v3.

```
NetIron(config-if-e100-1/3-vrid-13)# version v3
```

Syntax: [no] version <*v2 | v3*>

- VRRP v2 supports IPv4 environment
- VRRP v3 supports IPv4 and IPv6 environment

The default configuration is VRRP v2.

Configuring the Owner for IPv4

To configure the VRRP Owner router for IPv4, enter the following commands on the router.

```
NetIron1(config)# router vrrp
NetIron1(config)# interface ethernet 1/6
NetIron1(config-if-e10000-1/6)# ip address 192.53.5.1/24
NetIron1(config-if-e10000-1/6)# ip vrrp vrid 1
NetIron1(config-if-e10000-1/6-vrid-1)# owner
NetIron1(config-if-e10000-1/6-vrid-1)# ip-address 192.53.5.1
NetIron1(config-if-e10000-1/6-vrid-1)# activate
```

Syntax: [no] router vrrp

Syntax: [no] ip vrrp vrid <*num*>

Syntax: [no] owner [track-priority <*value*>**]**

Syntax: [no] activate

The <*num*> parameter specifies the virtual router ID.

The **track-priority** <*value*> parameter changes the track-port priority for this interface and VRID from the default (2) to a value from 1 through maximum VRID supported by the device.

Syntax: [no] ip-address <*ip-addr*>

The <*ip-addr*> parameter specifies the IPv4 address of the Owner router.

The IP address you assign to the Owner must be an IP address configured on an interface that belongs to the virtual router.

Refer to "Configuration rules and feature limitations for VRRP" on page 567 for additional requirements.

Configuring the Owner for IPv6

To configure the VRRP Owner router for IPv6, enter the following commands on the router.

```
NetIron1(config)# ipv6 router vrrp
NetIron1(config)# interface ethernet 1/6
NetIron1(config-if-e10000-1/6)# ipv6 address 3013::1/64
NetIron1(config-if-e10000-1/6)# ipv6 vrrp vrid 1
NetIron1(config-if-e10000-1/6-vrid-1)# owner
NetIron1(config-if-e10000-1/6-vrid-1)# ipv6-address 3013::1
NetIron1(config-if-e10000-1/6-vrid-1)# activate
```

Syntax: [**no**] **ipv6 router vrrp**

Syntax: [**no**] **ipv6 vrrp vrid** <*num*>

Syntax: [**no**] **ipv6-address** <*ipv6-addr*>

The <*num*> parameter specifies the virtual router ID.

The <*ipv6-addr*> parameter specifies the IPv6 address of the Owner router.

The IP address you assign to the Owner must be an IP address configured on an interface that belongs to the virtual router.

Refer to "Configuration rules and feature limitations for VRRP" on page 567 for additional requirements.

Configuring a Backup for IPv4

To configure the VRRP Backup router for IPv4, enter the following commands.

```
NetIron2(config)# router vrrp
NetIron2(config)# interface ethernet 1/5
NetIron2(config-if-e10000-1/5)# ip address 192.53.5.3/24
NetIron2(config-if-e10000-1/5)# ip vrrp vrid 1
NetIron2(config-if-e10000-1/5-vrid-1)# backup
NetIron2(config-if-1/5-vrid-1)# advertise backup
NetIron2(config-if-e10000-1/5-vrid-1)# ip-address 192.53.5.1
NetIron2(config-if-e10000-1/5-vrid-1)# activate
```

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

Syntax: [no] router vrrp

Syntax: [no] ip vrrp vrid <*num*>

Syntax: [no] backup [priority <*value*>**] [track-priority** <*value*>**]**

The <*num*> parameter specifies the virtual router ID.

The **priority** <*value*> parameter specifies the VRRP priority for this virtual router. You can specify a value from 3 through 254. The default is 100.

Enter a value from 3 through 254 for the **track-priority** <*value*> parameter if you want VRRP to monitor the state of the interface. The default is 100.

Syntax: [no] ip-address <*ip-addr*>

Refer to "Configuration rules and feature limitations for VRRP" on page 567 for additional requirements.

Configuring a Backup for IPv6

To configure the VRRP Backup router for IPv6, enter the following commands.

```
NetIron2(config)# ipv6 router vrrp
NetIron2(config)# interface ethernet 1/5
NetIron2(config-if-e10000-1/5)# ipv6 address 3013::3/64
NetIron2(config-if-e10000-1/5)# ipv6 vrrp vrid 1
NetIron2(config-if-e10000-1/5-vrid-1)# backup
NetIron2(config-if-1/5-vrid-1)# advertise backup
NetIron2(config-if-e10000-1/5-vrid-1)# ipv6-address 3013::1
NetIron2(config-if-e10000-1/5-vrid-1)# activate
```

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

Syntax: [**no**] **ipv6 router vrrp**

Syntax: [**no**] **ipv6-address** <*ipv6-addr*>

The <*num*> parameter specifies the virtual router ID.

The <*ipv6-addr*> parameter specifies the IPv6 address of the Backup router.

Refer to "Configuration rules and feature limitations for VRRP" on page 567 for additional requirements.

Configuration rules and feature limitations for VRRP

Consider the following rules when configuring VRRP:

- The interfaces of all routers in a virtual router must be in the same IP subnet:
- The IP addresses associated with the virtual router must already be configured on the router that will be the Owner router.
- The IP address for the virtual router must be on only one router.
- The Hello interval must be set to the same value on both the Owner and Backups for the virtual router.
- The Dead interval must be set to the same value on both the Owner and Backups for the virtual router.
- The track priority on a router must be lower than the router's VRRP priority. Also, the track priority on the Owner must be higher than the track priority on the Backups.
- The tracking-port configuration for IPv6 VRRP v3 is not allowed if the router is configured as the VRRP Owner.
- The priority configuration for IPv6 VRRP v3 is not allowed for Owner router. The Owner router's priority is always 255.
- Hitless switchover is not supported.
- The ping command is not supported for VRRP virtual IPv4 and IPv6 addresses.
- Mixed mode VRRP v2 and VRRP v3 is not supported in the same VRRP group.

Configuring parameters specific to VRRP-E

The following sections describe the configuration of the parameters specific to IPv4 and IPv6 VRRP-E.

Configuring IPv4 VRRP-E

VRRP-E is configured at the interface level. To implement a simple IPv4 VRRP-E configuration using all the default values, enter the following commands.

```
NetIron(config)# router vrrp-extended
NetIron(config)# interface ethernet 1/5
NetIron(config-if-e10000-1/5)# ip address 192.53.5.3/24
NetIron(config-if-e10000-1/5)# ip vrrp-extended vrid 1
NetIron(config-if-e10000-1/5-vrid-1)# backup priority 50 track-priority 10
NetIron(config-if-e10000-1/5-vrid-1)# ip-address 192.53.5.254
NetIron(config-if-e10000-1/5-vrid-1)# activate
```

Syntax: [no] **ip vrrp-extended vrid** <*vrid*>

Syntax: [no] **backup** [**priority** <*value*>] [**track-priority** <*value*>]

Syntax: [no] **ip-address** <*ip-addr*>

The <*vrid*> parameter specifies the virtual router ID.

The <*ip-addr*> parameter specifies the IPv4 address of the router.

Refer to the section "Authentication type" on page 569 for information on the **auth-type no-auth** | **simple-text-auth** <*auth-data*> parameters.

Also, refer to "Configuration rules and feature limitations for VRRP-E" on page 568 additional information on how to configure VRRP-E.

PowerConnect requires you to identify a VRRP-E router as a Backup before you can activate the virtual router. However, after you configure the virtual router, you can use the **backup** command to change its priority or track priority.

You also can use the **enable** command to activate the configuration. This command does the same thing as the **activate** command.

Configuring IPv6 VRRP-E

To implement a IPv6 VRRP-E configuration using all the default values, enter the following commands.

```
NetIron(config)# ipv6 router vrrp-extended
NetIron(config-ipv6-VRRP-E-router)# interface ethernet 1/5
NetIron(config-if-e10000-1/5)# ipv6 address 3013::2/64
NetIron(config-if-e10000-1/5)# ipv6 vrrp-extended vrid 1
NetIron(config-if-e10000-1/5-vrid-1)# backup priority 50 track-priority 10
NetIron(config-if-e10000-1/5-vrid-1)# ipv6-address 3013::99
NetIron(config-if-e10000-1/5-vrid-1)# activate
```

Syntax: ipv6 router vrrp-extended

Syntax: ipv6 vrrp-extended vrid <*vrid*>

Syntax: [no] ipv6-address <*ipv6-addr*>

The <*vrid*> parameter specifies the virtual router ID.

The <*ipv6-addr*> parameter specifies the IPv6 address of the router.

Configuration rules and feature limitations for VRRP-E

Consider the following rules when configuring VRRP-E:

- The interfaces of all routers in a virtual router must be in the same IP subnet.
- The IP address assigned to the virtual router cannot be configured on any of the PowerConnect devices.
- The Hello interval must be set to the same value on all the PowerConnect devices.
- The Dead interval must be set to the same value on all the PowerConnect devices.
- The track priority for a virtual router must be lower than the VRRP-E priority.
- The same VRID must not be used across IPv6 VRRP-E and IPv4 VRRP-E if they are in the same broadcast domain.
- Hitless switchover is not supported.

NOTE
If you disable VRRP-E, the PowerConnect removes all the configuration information for the disabled protocol from the running configuration. Moreover, when you save the configuration to the startup configuration after disabling the protocol, all configuration information for the disabled protocol is removed from the startup configuration.

Configuring additional VRRP and VRRP-E parameters

You can modify the following VRRP and VRRP-E parameters on each individual virtual router. These parameters apply to both protocols:

- Authentication type (if the interfaces on which you configure the virtual router use authentication)
- Backup priority
- Suppression of RIP advertisements on Backup routes for the backed up interface
- Hello interval
- Dead interval
- Backup Hello messages and message timer (Backup advertisement)
- Track port
- Track priority
- Backup preempt mode
- Master Router Abdication and Reinstatement
- VRRP-Extended Slow Start
- VRRP-Extended Scale Timer

Refer to "VRRP and VRRP-E parameters" on page 561 for a summary of the parameters and their defaults.

Authentication type

If the interfaces on which you configure the virtual router use authentication, the VRRP or VRRP-E packets on those interfaces also must use the same authentication. Dell's implementation of VRRP and VRRP-E supports the following authentication types:

- **No authentication** – The interfaces do not use authentication. This is the default for VRRP and VRRP-E.
- **Simple** – The interfaces use a simple text-string as a password in packets sent on the interface. If the interfaces use simple password authentication, the virtual router configured on the interfaces must use the same authentication type and the same password.

To configure the interface on Router1 for simple-password authentication using the password "ourpword", enter the following commands.

Configuring router 1

```
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ip vrrp auth-type simple-text-auth ourpword
```

Configuring router 2

```
Router2(config)# interface ethernet 1/5
Router2(config-if-e10000-1/5)# ip vrrp auth-type simple-text-auth ourpword
```

Syntax: [no] ip vrrp auth-type no-auth | simple-text-auth <*auth-data*>

The **auth-type no-auth** parameter indicates that the virtual router and the interface it is configured on do not use authentication.

The **auth-type simple-text-auth** <*auth-data*> parameter indicates that the virtual router and the interface it is configured on use a simple text password for authentication. The <*auth-data*> parameter is the password. If you use this parameter, make sure all interfaces on all the routers supporting this virtual router are configured for simple password authentication and use the same password.

NOTE
Authentication is not supported by VRRP v3.

Suppressingf RIP advertisements on backup routers for the backup up interface

Normally, a VRRP or VRRP-E backup includes route information for the virtual IP address in RIP advertisements. As a result, other routers receive multiple paths for the backup router and might sometimes unsuccessfully use the path to the backup router rather than the path to the Master.

You can prevent the backup routers from advertising route information for the interface on which they are defined by enabling suppression of the advertisements.

To suppress RIP advertisements for interface on which a backup router is defined in Router2, enter the following commands.

```
Router2(config)# router rip
Router2(config-rip-router)# use-vrrp-path
```

Syntax: [no] use-vrrp-path

The syntax is the same for VRRP and VRRP-E.

Hello interval

The Master periodically sends Hello messages to the Backups. The Backups use the Hello messages as verification that the Master is still on-line. If the Backup routers stop receiving the Hello messages for the period of time specified by the Dead interval, the Backup routers determine that the Master router is dead. At this point, the Backup router with the highest priority becomes the new Master router.

The default Dead interval is three times the Hello Interval plus one-half second. Generally, if you change the Hello interval, you also must change the Dead interval on the Backup routers.

To change the Hello interval on the Master to 10 seconds for VRRP v2 and VRRP-E v2, enter the following commands.

```
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ip vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# hello-interval 10
```

Syntax: [no] hello-interval <*value*>

The Hello interval can be from 1 through 84 seconds. The default is 1 second.

To change the Hello interval on the Master to 200 milliseconds for VRRP v3 and VRRP-E v3, enter the following commands.

```
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ip vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# hello-interval msec 200
```

Syntax: [**no**] **hello-interval** [**msec**] <*value*>

The Hello interval can be from 100 through 8400 milliseconds. The default is 1000 milliseconds.

The syntax is the same for VRRP and VRRP-E.

Dead interval

The Dead interval is the time a Backup waits for a Hello message from the Master before determining that the Master is dead. When Backups determine that the Master is dead, the Backup with the highest priority becomes the new Master. The dead interval is three times the default Hello interval (1 second) plus one-half second added by the router software. The software automatically adds one-half second to the Dead interval value you enter.

To change the Dead interval on a Backup to 30 seconds for VRRP v2 and VRRP-E v2, enter the following commands.

```
Router2(config)# interface ethernet 1/5
Router2(config-if-e10000-1/5)# ip vrrp vrid 1
Router2(config-if-e10000-1/5-vrid-1)# dead-interval 30
```

Syntax: [no] dead-interval <*value*>

The Dead interval can be from 1 through 84 seconds. The default is 3.5 seconds.

To change the Dead interval on a Backup to 600 milliseconds for VRRP v3 and VRRP-E v3, enter the following commands.

```
Router2(config)# interface ethernet 1/5
Router2(config-if-e10000-1/5)# ip vrrp vrid 1
Router2(config-if-e10000-1/5-vrid-1)# dead-interval msec 600
```

Syntax: [**no**] **dead-interval** [**msec**] <*value*>

The Dead interval can be from 100 through 8400 milliseconds. The default is 3500 milliseconds.

The syntax is the same for VRRP and VRRP-E.

Backup hello message state and interval

By default, Backup do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup to send Hello messages to the Master, enter the following commands.

```
NetIron(config)# router vrrp
NetIron(config)# interface ethernet 1/6
NetIron(config-if-e10000-1/6)# ip vrrp vrid 1
NetIron(config-if-e10000-1/6-vrid-1)# advertise backup
```

Syntax: [no] advertise backup

When you enable a Backup to send Hello messages, the Backup sends a Hello messages to the Master every 60 seconds by default. You can change the interval to be up to 3600 seconds. To do so, enter the following commands.

```
NetIron(config)# router vrrp
NetIron(config)# interface ethernet 1/6
NetIron(config-if-e10000-1/6)# ip vrrp vrid 1
NetIron(config-if-e10000-1/6-vrid-1)# backup-hello-interval 180
```

Syntax: [no] backup-hello-interval <*num*>

The <*num*> parameter specifies the message interval and can be from 60 through 3600 seconds. The default is 60 seconds.

The syntax is the same for VRRP and VRRP-E.

Track port

You can configure the virtual router to track the link state of interfaces on the PowerConnect. This capability is quite useful for tracking the state of the exit interface for the path for which the virtual router is providing redundancy. Refer to "Track ports and track priority" on page 557.

To configure 1/6 on Router1 to track interface 2/4, enter the following commands.

```
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ip vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
```

Syntax: [no] track-port ethernet <*slot*>/<*portnum*> **| ve** <*num*> **[priority** <*num*>**]**

The syntax is the same for VRRP and VRRP-E.

Track priority

If you configure a virtual router to track the link state of interfaces and one of the tracked interface goes down, the software changes the VRRP or VRRP-E priority of the virtual router:

- For VRRP, the software changes the priority of the virtual router to a track priority that is lower than that of the virtual router priority and lower than the priorities configured on the Backups. For example, if the virtual router priority is 100 and a tracked interface with track priority 60 goes down, the software changes the virtual router priority to 60.
- For VRRP-E, the software reduces the virtual router priority by the amount of the priority of the tracked interface that went down. For example, if the VRRP-E interface's priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VRRP-E interface's priority to 40. If another tracked interface goes down, the software reduces the virtual router's priority again, by the amount of the tracked interface's track priority.

The default track priority for a VRRP Owner is 2. The default track priority for Backups is 1.

You enter the track priority as a parameter with the **owner** or **backup** command. Refer to "Track port" on page 572.

Syntax: [no] owner [track-priority <*value*>**]**

Syntax: [no] backup [priority <*value*>**] [track-priority** <*value*>**]**

The syntax is the same for VRRP and VRRP-E.

Backup preempt

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the virtual router. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the virtual router.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, since the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

NOTE
In VRRP, regardless of the setting for the preempt parameter, the Owner always returns to be the Master when it comes back online.

To disable preemption on a Backup, enter the following commands.

```
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ip vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# non-preempt-mode
```

Syntax: [no] non-preempt-mode

The syntax is the same for VRRP and VRRP-E.

Master router abdication and reinstatement

To change the Master's priority, enter the following commands.

```
NetIron(config)# interface ethernet 1/6
NetIron(config-if-e10000-1/6)# ip vrrp vrid 1
NetIron(config-if-e10000-1/6-vrid-1)# owner priority 99
```

Syntax: [no] owner priority | track-priority <*num*>

The <*num*> parameter specifies the new priority and can be a number from 1 through 254.

When you press Enter, the software changes the priority of the Master to the specified priority. If the new priority is lower than at least one Backup's priority for the same virtual router, the Backup takes over and becomes the new Master until the next software reload or system reset.

To verify the change, enter the following command from any level of the CLI.

```
NetIron(config-if-e10000-1/6-vrid-1)# show ip vrrp
Total number of VRRP routers defined: 1
Interface ethernet 1/6
auth-type no authentication
VRID 1
state backup
administrative-status enabled
mode owner
```

```
priority 99
current priority 99
hello-interval 1 sec
ip-address 192.53.5.1
backup routers 192.53.5.2
```

This example shows that even though this PowerConnect is the Owner of the virtual router ("mode owner"), the PowerConnect's priority for the virtual router is only 99 and the state is now "backup" instead of "active". In addition, the administrative status is "enabled".

To change the Master's priority back to the default Owner priority 255, enter "no" followed by the command you entered to change the priority. For example, to change the priority of a VRRP Owner back to 255 from 99, enter the following command.

```
NetIron(config-if-e10000-1/6-vrid-1)# no owner priority 99
```

You cannot set the priority to 255 using the **owner priority** command.

VRRP-extended slow start

In a VRRP-E configuration, if a Master router goes down, the Backup router with the highest priority takes over after expiration of the dead interval. When the original Master router comes back up again, it takes over from the Backup router (which became the Master router when the original Master router went down). By default, this transition from Backup router back to Master router takes place after expiration of the dead interval.

You can configure the VRRP-E slow start timer feature, which causes a specified amount of time to elapse between the time the original Master router is restored and when it takes over from the Backup router (This range is currently set to between 1-60 seconds). This interval allows time for OSPF convergence when the Master is restored.

To set the VRRP-E slow start timer to 30 seconds, enter the following command.

```
NetIron(config)# router vrrp-extended
NetIron(config-vrrpe-router)# slow-start 30
```

Syntax: [no] slow-start <*seconds*>

When the VRRP-E slow start timer is enabled, if the Master router goes down, the Backup router takes over after expiration of the dead interval. If the original Master router subsequently comes back up again, the amount of time specified by the VRRP-E slow start timer elapses (in this example, 30 seconds) before the original Master router takes over from the Backup router (which became the Master router when the original Master router went down).

In the event that no other routers are currently Master, the router will immediately (after the dead-interval) become Master.

VRRP-extended scale timer

This feature allows you to increase timing sensitivity across all configured or default VRRP-Extended timers. When this command is used, all configured or default VRRP-Extended timers are divided by the value set in the command. For example: a value of 10 divides the timers by a factor of 10. Configuring a value of 10 in a network with all VRRP-Extended values set to their defaults would cause VRRP-Extended instances to send packets every 100ms (instead of every 1

second) and the backup advertisement interval of 60 seconds would be modified to 6 seconds. All other timers would be divided likewise. This would allow VRRP-Extended instances to converge within a second in the event of a VRRP-Extended master failure (this is since the default dead timer would be 300 ms).

To scale all VRRP-Extended timers by 10, use the following command.

```
NetIron(config)# scale-timer vrrp-extended 10
```

Syntax: **[no] scale-timer vrrp-extended** <*scale-factor*>

The <*scale-factor*> variable is the number that all VRRP-Extended timers values are divided by. Values can be set from 1 through 10.

NOTE
Increased timing sensitivity as a result of using this command could cause protocol flaps during times of network congestion.

NOTE
This command is not supported in VRRP v2 and VRRP v3.

Displaying VRRP and VRRP-E information for IPv4

You can display the following information for IPv4 VRRP or VRRP-E:

- "Displaying summary information"
- "Displaying detailed information"
- "Displaying statistics"
- "Displaying configuration information for VRRP and VRRP-E"

Displaying summary information

To display summary information for IPv4 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
NetIron(config)# show ip vrrp brief
Total number of VRRP routers defined: 2
Flags Codes - P:Preempt 2:V2 3:V3 S:Short-Path-Fwd

Inte- VRID Current  Flags  State  Master IP       Backup IP       Virtual IP
rface      Priority               Address         Address         Address
-------------------------------------------------------------------------------
1/1   10   255      P2-    Master Local           Unknown         30.30.30.2
1/3   13   100      P2-    Master Local           Unknown         10.13.13.3
```

Syntax: **show ip vrrp [brief | ethernet** <*slot*>/<*portnum*> **| statistics | ve** <*num*> **| vrid** <*id*>**]**

Syntax: **show ip vrrp-extended [brief | ethernet** <*slot*>/<*portnum*> **| statistics | ve** <*num*> **| vrid** <*id*>**]**

The **brief** parameter displays the summary information. If you do not use this parameter, detailed information is displayed instead. Refer to "Displaying detailed information" on page 577.

The **ethernet** <*slot*>/<*portnum*> parameter specifies an Ethernet port. If you use this parameter, the command displays IPv4 VRRP or VRRP-E information only for the specified port.

The **ve** <*num*> parameter specifies a virtual interface. If you use this parameter, the command displays IPv4 VRRP or VRRP-E information only for the specified virtual interface.

The **vrid** <*id*> parameter specifies a virtual router ID. If you use this parameter, the command displays IPv4 VRRP or VRRP-E information only for the specified virtual router.

The **statistics** parameter displays statistics. Refer to "Displaying statistics" on page 580.

This display shows the following information.

TABLE 100 CLI display of VRRP or VRRP-E summary information

This field...	Displays...
Total number of VRRP (or VRRP-Extended) routers defined	The total number of virtual routers configured on this PowerConnect. **NOTE:** The total applies only to the protocol the PowerConnect is running. For example, if the PowerConnect is running VRRP-E, the total applies only to VRRP-E routers.
Interface	The interface on which VRRP or VRRP-E is configured. If VRRP or VRRP-E is configured on multiple interfaces, information for each interface is listed separately.
VRID	The ID of the virtual router configured on this interface. If multiple virtual routers are configured on the interface, information for each virtual router is listed in a separate row.
Current Priority	The current VRRP or VRRP-E priority of this PowerConnect for the virtual router.
Flags	Whether the backup preempt mode is enabled. If the backup preempt mode is enabled, this field contains a "P". If the mode is disabled, this field is blank. P:Preempt 2:V2 3:V3 2: imples VRRP Version2 3 implies VRRP Version3.
Short-Path-Fwd	Displays information about whether VRRP-E Extension for Server Virtualization is enabled or disabled.
State	This PowerConnect's VRRP or VRRP-E state for the virtual router. The state can be one of the following: • Init – The virtual router is not enabled (activated). If the state remains Init after you activate the virtual router, make sure that the virtual router is also configured on the other routers and that the routers can communicate with each other. **NOTE:** If the state is Init and the mode is incomplete, make sure you have specified the IP address for the virtual router. • Backup – This PowerConnect is a Backup for the virtual router. • Master – This PowerConnect is the Master for the virtual router.
Master addr	The IP address of the router interface that is currently the Master for the virtual router.
Backup addr	The IP addresses of the router interfaces that are currently Backups for the virtual router.
VIP	The virtual IP address that is being backed up by the virtual router.

Displaying detailed information

To display detailed information for IPv4 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
NetIron(config)# show ip vrrp-extended
Total number of vrrp-extended routers defined: 1
Interface v10
----------------
auth-type no authentication
VRID 10 (index 1)
 interface v10
 state backup
 administrative-status enabled
 mode non-owner(backup)
 virtual mac 02e0.52a0.c00a
 priority 50
 current priority 50
 track-priority 5
 hello-interval 1 sec
 backup hello-interval 60 sec
 advertise backup disabled
 dead-interval 0 sec
 current dead-interval 3.6 sec
 preempt-mode true
 virtual ip address 10.10.10.254
master router 10.10.10.4 expires in 3.1 sec
short-path-forwarding enabled
```

Syntax: show ip vrrp

Syntax: show ip vrrp-extended

This display shows the following information.

TABLE 101 CLI display of VRRP or VRRP-E detailed information

This field...	Displays...
Total number of VRRP (or VRRP-Extended) routers defined	The total number of virtual routers configured on this PowerConnect. **NOTE:** The total applies only to the protocol the PowerConnect is running. For example, if the PowerConnect is running VRRP-E, the total applies only to VRRP-E routers.
Interface parameters	
Interface	The interface on which VRRP or VRRP-E is configured. If VRRP or VRRP-E is configured on multiple interfaces, information for each interface is listed separately.
auth-type	The authentication type enabled on the interface.
Virtual Router parameters	
VRID	The virtual router configured on this interface. If multiple virtual routers are configured on the interface, information for each virtual router is listed separately.

TABLE 101 CLI display of VRRP or VRRP-E detailed information (Continued)

This field...	Displays...
state	This PowerConnect's VRRP or VRRP-E state for the virtual router. The state can be one of the following: • initialize – The virtual router is not enabled (activated). If the state remains "initialize" after you activate the virtual router, make sure that the virtual router is also configured on the other routers and that the routers can communicate with each other. **NOTE:** If the state is "initialize" and the mode is incomplete, make sure you have specified the IP address for the virtual router. • backup – This PowerConnect is a Backup for the virtual router. • master – This PowerConnect is the Master for the virtual router.
administrative-status	The administrative status of the virtual router. The administrative status can be one of the following: • disabled – The virtual router is configured on the interface but VRRP or VRRP-E has not been activated on the interface. • enabled – VRRP or VRRP-E has been activated on the interface.
mode	Indicates whether the PowerConnect is the Owner or a Backup for the virtual router. **NOTE:** If "incomplete" appears after the mode, configuration for this virtual router is incomplete. For example, you might not have configured the virtual IP address that is being backup up by the virtual router. This field applies only to VRRP. All PowerConnect devices configured for VRRP-E are Backups.
virtual MAC	The virtual IP MAC address that this virtual router is backing up.
priority	The device's preferability for becoming the Master for the virtual router. During negotiation, the router with the highest priority becomes the Master. If two or more devices are tied with the highest priority, the Backup interface with the highest IP address becomes the active router for the virtual router.
current priority	The current VRRP or VRRP-E priority of this PowerConnect for the virtual router. The current priority can differ from the configured priority (refer the previous row) for the following reasons: • The virtual router is still in the initialization stage and has not become a Master or Backup yet. In this case, the current priority is 0. • The virtual router is configured with track ports and the link on a tracked interface has gone down. Refer to "Track ports and track priority" on page 557.
track priority	VRRP-E priority value assigned to the tracked port.
hello-interval	The number of seconds between Hello messages from the Master to the Backups for a given virtual router.
backup hello-interval	The number of seconds between Hello messages from a Backup to the Master.

TABLE 101 CLI display of VRRP or VRRP-E detailed information (Continued)

This field...	Displays...
advertise backup	The IP addresses of Backups that have advertised themselves to this PowerConnect by sending Hello messages. **NOTE:** Hello messages from Backups are disabled by default. You must enable the Hello messages on the Backup for the Backup to advertise itself to the current Master. Refer to "Hello interval" on page 570.
dead-interval	The configured value for the dead interval. The dead interval is the number of seconds a Backup waits for a Hello message from the Master for the virtual router before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the virtual router. **NOTE:** If the value is 0, then you have not configured this parameter. This field does not apply to VRRP Owners.
current dead-interval	The current value of the dead interval. This is the value actually in use by this interface for the virtual router. **NOTE:** This field does not apply to VRRP Owners.
preempt-mode	Whether the backup preempt mode is enabled. **NOTE:** This field does not apply to VRRP Owners.
virtual ip address	The virtual IP addresses that this virtual router is backing up.
backup router <*ip-addr*> expires in <*time*>	The IP addresses of Backups that have advertised themselves to this Master by sending Hello messages. The <*time*> value indicates how long before the Backup expires. A Backup expires if you disable the advertise backup option on the Backup or the Backup becomes unavailable. Otherwise, the Backup's next Hello message arrives before the Backup expires. The Hello message resets the expiration timer. An expired Backup does not necessarily affect the Master. However, if you have not disabled the advertise backup option on the Backup, then the expiration may indicate a problem with the Backup. **NOTE:** This field applies only when Hello messages are enabled on the Backups (using the advertise backup option).
next hello sent in <*time*>	How long until the Backup sends its next Hello message. **NOTE:** This field applies only when this PowerConnect is the Master and the Backup is configured to send Hello messages (the advertise backup option is enabled).
master router <*ip-addr*> expires in <*time*>	The IP address of the Master and the amount of time until the Master's dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this PowerConnect itself will become the Master. **NOTE:** This field applies only when this PowerConnect is a Backup.

TABLE 101 CLI display of VRRP or VRRP-E detailed information (Continued)

This field...	Displays...
track port	The interfaces that the virtual router's interface is tracking. If the link for a tracked interface goes down, the VRRP or VRRP-E priority of the virtual router interface is changed, causing the devices to renegotiate for Master. **NOTE:** This field is displayed only if track interfaces are configured for this virtual router.
short-path-forwarding	Displays information about whether VRRP-E Extension for Server Virtualization is enabled or disabled.

Displaying statistics

To display IPv4 VRRP or VRRP-E statistics, enter the following command.

```
NetIron# show ip vrrp-extended statistics

Global VRRP-Extended statistics
-------------------------------
- received vrrp-extended packets with checksum errors = 0
- received vrrp-extended packets with invalid version number = 0
- received vrrp-extended packets with unknown or inactive vrid = 1480

Interface v10
----------------
VRID 1
- number of transitions to backup state = 1
- number of transitions to master state = 1
- total number of vrrp-extended packets received = 0
  . received backup advertisements = 0
  . received packets with zero priority = 0
  . received packets with invalid type = 0
  . received packets with invalid authentication type = 0
  . received packets with authentication type mismatch = 0
  . received packets with authentication failures = 0
  . received packets dropped by owner = 0
  . received packets with ip ttl errors = 0
  . received packets with ip address mismatch = 0
  . received packets with advertisement interval mismatch = 0
  . received packets with invalid length = 0
- total number of vrrp-extended packets sent = 2004
  . sent backup advertisements = 0
  . sent packets with zero priority = 0
- received arp packets dropped = 0
- received proxy arp packets dropped = 0
- received ip packets dropped = 0
```

Syntax: show ip vrrp statistics

Syntax: show ip vrrp-extended statistics

The **statistics** parameter displays the following information.

The *received vrrp packets with checksum errors* shows the number of packets that is contained in checksum errors.

The *received vrrp packets with invalid version number* shows the number of packets with invalid versions.

The *received vrrp packets with unknown or inactive vrid* shows the number of packets that contain virtual routers that are not configured on the device or its interface

Displaying VRRP and VRRP-E information for IPv6

You can display the following information for IPv6 VRRP or VRRP-E:

- "Displaying summary information"
- "Displaying detailed information"
- "Displaying statistics"
- "Displaying configuration information for VRRP and VRRP-E"

Displaying summary information

To display summary information for IPv6 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
NetIron(config)# show ipv6 vrrp-extended brief
Total number of VRRP routers defined: 1
Flags Codes - P:Preempt 2:V2 3:V3 S:Short-Path-Fwd

Intf  VRID CurrPrio Flags State  Master-IPv6 Backup-IPv6 Virtual-IPv6
                                 -Address    -Address    -Address
-------------------------------------------------------------------
1/3   23   100      P3-   Master   Local     3013::2       3013::99
```

Syntax: **show ipv6 vrrp** [**brief** | **ethernet** <*slot*>/<*portnum*> | **statistics** | **ve** <*num*> | **vrid** <*id*>]

Syntax: **show ipv6 vrrp-extended** [**brief** | **ethernet** <*slot*>/<*portnum*> | **statistics** | **ve** <*num*> | **vrid** <*id*>]

The **brief** parameter displays the summary information. If you do not use this parameter, detailed information is displayed instead. Refer to "Displaying detailed information" on page 577.

The **ethernet** <*slot*>/<*portnum*> parameter specifies an Ethernet port. If you use this parameter, the command displays IPv6 VRRP or VRRP-E information only for the specified port.

The **ve** <*num*> parameter specifies a virtual interface. If you use this parameter, the command displays IPv6 VRRP or VRRP-E information only for the specified virtual interface.

The **vrid** <*id*> parameter specifies a virtual router ID. If you use this parameter, the command displays IPv6 VRRP or VRRP-E information only for the specified virtual router.

The **statistics** parameter displays statistics. Refer to "Displaying statistics" on page 580.

Displaying detailed information

To display detailed information for IPv6 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
NetIron(config)# show ipv6 vrrp

Total number of VRRP routers defined: 1

Interface 1/3
----------------
auth-type no authentication

VRID 13 (index 2)
 interface 1/3
 state master
 administrative-status enabled
 version v3
 mode non-owner(backup)
 virtual mac 0000.5e00.0217
 priority 100
 current priority 100
 track-priority 1
 hello-interval 1000 ms
 backup hello-interval 60000 ms
 advertise backup disabled
 dead-interval 3000 ms
 preempt-mode true
 ipv6-address 3013::1
 next hello sent in 700 ms
 short-path-forwarding disabled
```

Syntax: show ipv6 vrrp

Syntax: show ipv6 vrrp-extended

Displaying statistics

To display IPv6 VRRP or VRRP-E statistics, enter the following command.

```
NetIron# show ipv6 vrrp statistics

Global IPv6 VRRP statistics
------------------------------
- received vrrp packets with checksum errors = 0
- received vrrp packets with invalid version number = 0
- received vrrp packets with unknown or inactive vrid = 0
Interface 1/3
----------------
VRID 13
- number of transitions to backup state = 1
- number of transitions to master state = 1
- total number of vrrp packets received = 0
. received backup advertisements = 19
. received packets with zero priority = 0
. received packets with invalid type = 0
. received packets with invalid authentication type = 0
. received packets with authentication type mismatch = 0
. received packets with authentication failures = 0
```

```
. received packets dropped by owner = 0
. received packets with ttl errors = 0
. received packets with ipv6 address mismatch = 0
. received packets with advertisement interval mismatch = 0
. received packets with invalid length = 0
- total number of vrrp packets sent = 1175
. sent backup advertisements = 0
. sent packets with zero priority = 0
- received neighbor solicitation packets dropped = 0
- received proxy neighbor solicitation packets dropped = 0
- received ipv6 packets dropped = 0
```

Syntax: show ipv6 vrrp statistics

Syntax: show ipv6 vrrp-extended statistics

Displaying configuration information for VRRP and VRRP-E

To display the current configuration information for IPv4 VRRP or VRRP-E and IPv6 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
NetIron(config-if-e10000-1/3)# show run
…
router vrrp
…
interface ethernet 1/1
 port-name Port111
 enable
 ip address 30.30.30.2/28
 ip vrrp vrid 10
  owner
  ip-address 30.30.30.2
  activate
!
…
interface ethernet 1/3
 enable
 ip address 10.13.13.2/24
 ip vrrp vrid 13
  version v3
  backup
  ip-address 10.13.13.3
  activate
…
```

Syntax: show run

Clearing VRRP or VRRP-E statistics

To clear IPv4 VRRP or VRRP-E statistics, enter the following command at the Privileged EXEC level or any configuration level of the CLI.

```
NetIron(config)# clear ip vrrp statistics
```

Syntax: clear ip vrrp statistics

Syntax: clear ip vrrp-extended statistics

To clear IPv6 VRRP or VRRP-E statistics, enter the following command at the Privileged EXEC level or any configuration level of the CLI.

```
NetIron(config)# clear ipv6 vrrp statistics
```

Syntax: clear ipv6 vrrp statistics

Syntax: clear ipv6 vrrp-extended statistics

Configuration examples

The following sections contain the CLI commands options for implementing the VRRP and VRRP-E configurations shown in Figure 106 on page 553 and Figure 107 on page 558.

VRRP example for IPv4

To implement the IPv4 VRRP configuration shown in Figure 106 on page 553, enter the following commands.

Configuring router1

To configure VRRP Router1, enter the following commands.

```
Router1(config)# router vrrp
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ip address 192.53.5.1
Router1(config-if-e10000-1/6)# ip vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# owner track-priority 20
Router1(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
Router1(config-if-e10000-1/6-vrid-1)# ip-address 192.53.5.1
Router1(config-if-e10000-1/6-vrid-1)# activate
```

NOTE
When you configure the Master (Owner), the address you enter with the **ip-address** command must already be configured on the interface.

The **ip vrrp owner** command specifies that this router owns the IP address you are associating with the virtual router. Since this router owns the IP address, this router is the default Master router and its VRRP priority is thus 255.

Configuring router2

To configure Router2 in Figure 106 on page 553 after enabling VRRP, enter the following commands.

```
Router2(config)# router vrrp
Router2(config)# interface ethernet 1/5
Router2(config-if-e10000-1/5)# ip address 192.53.5.3
Router2(config-if-e10000-1/5)# ip vrrp vrid 1
Router2(config-if-e10000-1/5-vrid-1)# backup priority 100 track-priority 19
Router2(config-if-e10000-1/5-vrid-1)# track-port ethernet 3/2
Router2(config-if-e10000-1/5-vrid-1)# ip-address 192.53.5.1
Router2(config-if-e10000-1/5-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRP Backup for virtual router VRID1. The IP address entered with the **ip-address** command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the virtual router Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this virtual router on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE
When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet as the address associated with the virtual router by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP priority in relation to the other VRRP routers in this virtual router. The **track-priority** parameter specifies the new VRRP priority that the router receives for this virtual router if the interface goes down. Refer to "Track ports and track priority" on page 557.

The **activate** command activates the virtual router configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP configuration.

Syntax: router vrrp

Syntax: ip vrrp vrid <*vrid*>

Syntax: owner [track-priority <*value*>**]**

Syntax: backup [priority <*value*>**] [track-priority** <*value*>**]**

Syntax: track-port ethernet <*slot*>/<*portnum*> **ve** <*num*>

Syntax: ip-address <*ip-addr*>

Syntax: activate

VRRP example for IPv6

To implement the VRRP configuration for IPv6, enter the following commands.

Configuring router1

To configure VRRP Router1, enter the following commands.

```
Router1(config)# ipv6 router vrrp
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ipv6 address 1414:1414:1414::1/64
Router1(config-if-e10000-1/6)# ipv6 vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# owner track-priority 20
Router1(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
Router1(config-if-e10000-1/6-vrid-1)# ipv6-address 1414:1414:1414::1/64
Router1(config-if-e10000-1/6-vrid-1)# activate
```

NOTE
When you configure the Master (Owner), the address you enter with the **ipv6-address** command must already be configured on the interface.

The **ipv6 vrrp owner** command specifies that this router owns the IP address you are associating with the virtual router. Since this router owns the IP address, this router is the default Master router and its VRRP priority is thus 255.

Configuring router2

To configure VRRP Router2, enter the following commands.

```
Router2(config)# ipv6 router vrrp
Router2(config)# interface ethernet 1/5
Router2(config-if-e10000-1/5)# ipv6 address 1414:1414:1414::2/64
Router2(config-if-e10000-1/5)# ipv6 vrrp vrid 1
Router2(config-if-e10000-1/5-vrid-1)# backup priority 100 track-priority 19
Router2(config-if-e10000-1/5-vrid-1)# track-port ethernet 3/2
Router2(config-if-e10000-1/5-vrid-1)# ipv6-address 1414:1414:1414::1/64
Router2(config-if-e10000-1/5-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRP Backup for virtual router VRID1. The IP address entered with the **ipv6-address** command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the virtual router Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this virtual router on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE
When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet as the address associated with the virtual router by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP priority in relation to the other VRRP routers in this virtual router. The **track-priority** parameter specifies the new VRRP priority that the router receives for this virtual router if the interface goes down.

The **activate** command activates the virtual router configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP configuration.

Syntax: ipv6 router vrrp

Syntax: ipv6 vrrp vrid <*vrid*>

Syntax: owner [**track-priority** <*value*>]

Syntax: **backup** [**priority** <*value*>] [**track-priority** <*value*>]

Syntax: **track-port ethernet** <*slot*>/<*portnum*> **ve** <*num*>

Syntax: **ipv6-address** <*ip-addr*>

Syntax: **activate**

VRRP-E example for IPv4

To implement the IPv4 VRRP-E configuration shown in Figure 107 on page 558, configure the VRRP Routers as shown in the following sections.

Configuring router1

To configure VRRP Router1 in Figure 107 on page 558, enter the following commands.

```
Router1(config)# router vrrp-extended
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ip address 192.53.5.2/24
Router1(config-if-e10000-1/6)# ip vrrp-extended vrid 1
Router1(config-if-e10000-1/6-vrid-1)# backup priority 110 track-priority 20
Router1(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
Router1(config-if-e10000-1/6-vrid-1)# ip-address 192.53.5.254
Router1(config-if-e10000-1/6-vrid-1)# activate
VRRP router 1 for this interface is activating
Router1(config-if-e10000-1/6-vrid-1)# exit
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ip vrrp-extended vrid 2
Router1(config-if-e10000-1/6-vrid-1)# backup priority 100 track-priority 20
Router1(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
Router1(config-if-e10000-1/6-vrid-1)# ip-address 192.53.5.253
Router1(config-if-e10000-1/6-vrid-1)# activate
VRRP router 2 for this interface is activating
```

NOTE
The address you enter with the **ip-address** command cannot be the same as a real IP address configured on the interface.

Configuring router2

To configure Router2, enter the following commands.

```
Router2(config)# router vrrp-extended
Router2(config)# interface ethernet 5/1
Router2(config-if-e10000-5/1)# ip address 192.53.5.3/24
Router2(config-if-e10000-5/1)# ip vrrp-extended vrid 1
Router2(config-if-e10000-5/1-vrid-1)# backup priority 100 track-priority 20
Router2(config-if-e10000-5/1-vrid-1)# track-port ethernet 3/2
Router2(config-if-e10000-5/1-vrid-1)# ip-address 192.53.5.254
Router2(config-if-e10000-5/1-vrid-1)# activate
Router2(config-if-e10000-5/1-vrid-1)# exit
Router2(config)# interface ethernet 5/1
Router2(config-if-e10000-5/1)# ip vrrp-extended vrid 2
Router2(config-if-e10000-5/1-vrid-1)# backup priority 110 track-priority 20
Router2(config-if-e10000-5/1-vrid-1)# track-port ethernet 2/4
Router2(config-if-e10000-5/1-vrid-1)# ip-address 192.53.5.253
Router2(config-if-e10000-5/1-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRP-E Backup for virtual router VRID1. The IP address entered with the **ip-address** VRRP-E command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the virtual router Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this virtual router on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE
When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet as the address associated with the virtual router by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP-E priority in relation to the other VRRP-E routers in this virtual router. The **track-priority** parameter specifies the new VRRP-E priority that the router receives for this virtual router if the interface goes down. Refer to "Track ports and track priority" on page 557.

The **activate** command activates the virtual router configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP-E configuration. Alternatively, you can use the **enable** command. The **activate** and **enable** commands do the same thing.

Syntax: [no] **router vrrp-extended**

Syntax: [no] **ip vrrp-extended vrid** <*vrid*>

Syntax: [no] **backup** [**priority** <*value*>] [**track-priority** <*value*>]

Syntax: [no] **track-port ethernet** <*slot*>/<*portnum*> **ve** <*num*>

Syntax: [no] **ip-address** <*ip-addr*>

Syntax: [no] **activate**

VRRP-E example for IPv6

To implement the IPv6 VRRP-E configuration, configure the VRRP Routers as shown in the following sections.

Configuring router1

To configure VRRP Router1, enter the following commands.

```
Router1(config)# ipv6 router vrrp-extended
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ipv6 address 1414:1414:1414::3/64
Router1(config-if-e10000-1/6)# ipv6 vrrp-extended vrid 1
Router1(config-if-e10000-1/6-vrid-1)# backup priority 110 track-priority 20
Router1(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
Router1(config-if-e10000-1/6-vrid-1)# ipv6-address 1414:1414:1414::45
Router1(config-if-e10000-1/6-vrid-1)# activate
VRRP router 1 for this interface is activating
Router1(config-if-e10000-1/6-vrid-1)# exit
Router1(config)# interface ethernet 1/6
Router1(config-if-e10000-1/6)# ipv6 vrrp-extended vrid 2
Router1(config-if-e10000-1/6-vrid-1)# backup priority 100 track-priority 20
```

```
Router1(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
Router1(config-if-e10000-1/6-vrid-1)# ipv6-address 1414:1414:1414::44
Router1(config-if-e10000-1/6-vrid-1)# activate
VRRP router 2 for this interface is activating
```

NOTE
The address you enter with the **ipv6-address** command cannot be the same as a real IP address configured on the interface.

Configuring router2

To configure Router2, enter the following commands.

```
Router2(config)# ipv6 router vrrp-extended
Router2(config)# interface ethernet 5/1
Router2(config-if-e10000-5/1)# ipv6 address 1414:1414:1414::4/64
Router2(config-if-e10000-5/1)# ipv6 vrrp-extended vrid 1
Router2(config-if-e10000-5/1-vrid-1)# backup priority 100 track-priority 20
Router2(config-if-e10000-5/1-vrid-1)# track-port ethernet 3/2
Router2(config-if-e10000-5/1-vrid-1)# ipv6-address 1414:1414:1414::45
Router2(config-if-e10000-5/1-vrid-1)# activate
Router2(config-if-e10000-5/1-vrid-1)# exit
Router2(config)# interface ethernet 5/1
Router2(config-if-e10000-5/1)# ipv6 vrrp-extended vrid 2
Router2(config-if-e10000-5/1-vrid-1)# backup priority 110 track-priority 20
Router2(config-if-e10000-5/1-vrid-1)# track-port ethernet 2/4
Router2(config-if-e10000-5/1-vrid-1)# ipv6-address 1414:1414:1414::44
Router2(config-if-e10000-5/1-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRP-E Backup for virtual router VRID1. The IP address entered with the **ipv6-address** VRRP-E command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the virtual router Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this virtual router on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE
When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet as the address associated with the virtual router by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP-E priority in relation to the other VRRP-E routers in this virtual router. The **track-priority** parameter specifies the new VRRP-E priority that the router receives for this virtual router if the interface goes down.

The **activate** command activates the virtual router configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP-E configuration. Alternatively, you can use the **enable** command. The **activate** and **enable** commands do the same thing.

Syntax: [no] ipv6 router vrrp-extended

Syntax: [no] ipv6 vrrp-extended vrid <*vrid*>

Syntax: [no] backup [priority <*value*>**] [track-priority** <*value*>**]**

Syntax: [no] track-port ethernet <*slot*>/<*portnum*> **ve** <*num*>

Syntax: [no] ipv6-address <*ip-addr*>

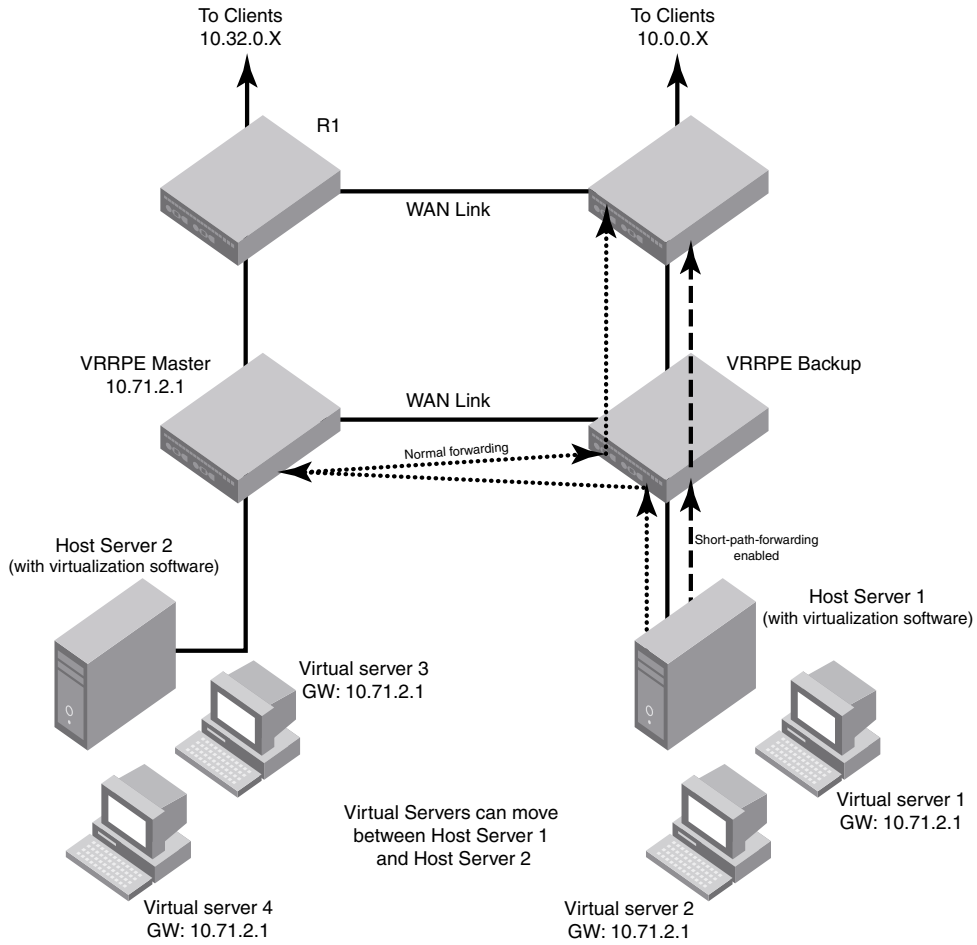
Syntax: [no] activate

VRRP-E Extension for Server Virtualization

VRRP-E is enhanced with the VRRP-E extension for Server Virtualization feature so that the PowerConnect attempts to bypass the VRRP-E master router and directly forward packets to their destination through interfaces on the Backup router.

Figure 108 shows an example of VRRP-E Extension for Server Virtualization. As shown, the virtual servers are dynamically moved between Host Server 1 and Host Server 2. Each time the virtual server is activated, it can be on a different Host Server, and sometimes the traffic crosses the WAN two times before it reaches the client. For example, in the VRRP-E implementation (without VRRP-E Extension for Server Virtualization), traffic from virtual server 1 to the client at 10.0.0.X was switched to the VRRP-E master router, then routed back to VRRP-E Backup router, and then routed to the client (the normal forwarding path, dotted lines).

FIGURE 108 Short path forwarding



VRRP-E Extension for server virtualization configuration example

Under the VRRP-E VRID configuration level, there is an option to enable short-path-forwarding.

To enable **short-path-forwarding,** enter the following commands.

```
NetIron(config)# router vrrp-extended
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ip address 10.10.10.25/24
NetIron(config-vif-10)# ip vrrp-extended vrid 10
NetIron(config-vif-10-vrid-10)# backup priority 50
NetIron(config-vif-10-vrid-10)# ip-address 10.10.10.254
NetIron(config-vif-10-vrid-10)# short-path-forwarding
NetIron(config-vif-10-vrid-10)# activate
```

Syntax: [no] short-path-forwarding

Packets from the local subnet of the virtual IP address

If VRRP-E Extension for Server Virtualization is enabled, any packets coming from the local subnet of the virtual IP address will be routed to the VRRP-E master router. This is for the routes whose next-hop gateway is the master router at the Backup router. These routes are routed to the WAN instead of switching them to the master router. The new behavior includes all the packets sent to the virtual IP address that were intended for the master router, such as Telnet, ping and traceroute packets. With VRRP-E Extension for Server Virtualization enabled these packets are now routed instead of switched. Traceroute output will show one extra hop for the source IP subnet that displays the Backup router interface IP address.

The following is an example of the traceroute command output with VRRP-E Extension for Server Virtualization enabled.

```
C:\ >traceroute 10.10.10.254
Tracing route to 10.10.10.254 over a maximum of 30 hops
1    <1 ms    <1 ms    <1 ms  10.10.10.25
 2    <1 ms    <1 ms    <1 ms  10.10.10.254
Trace complete.
```

The following is an example of the traceroute command output without VRRP-E Extension for Server Virtualization enabled:

```
C:\ >traceroute 10.10.10.254
Tracing route to 10.10.10.254 over a maximum of 30 hops
1     8 ms     8 ms     8 ms  10.10.10.254
Trace complete.
```

IPv4 VRF support

VRRP-E Extension for Server Virtualization supports IPv4 VRF forwarding.

Configuration considerations

Since the VRRP-E Extension for Server Virtualization enabled port will route all the traffic sent to the VRRP-E MAC, ACL and PBR features configured on the port will be applied to this traffic.

VRRP-E Extension for Server Virtualization can be dynamically enabled or disabled before or after VRRP-E is activated. No system reload is required.

Although it is not required, it is recommended that interfaces on different routers with the same VRID have the same short-path-forwarding configuration. This will ensure that the short-path forwarding behavior is still retained after the failover. Different VRIDs can be configured differently.

Chapter

Multi-Chassis Trunking (MCT) 18

The Multi-Chassis Trunking (MCT) features are supported by the NetIron MLX Series devices.

- Multi-Chassis Trunking (MCT)
- Cluster operation features
- Multi-Chassis Trunk (MCT) for VRRP or VRRP-E

About Multi-Chassis Trunk (MCT)

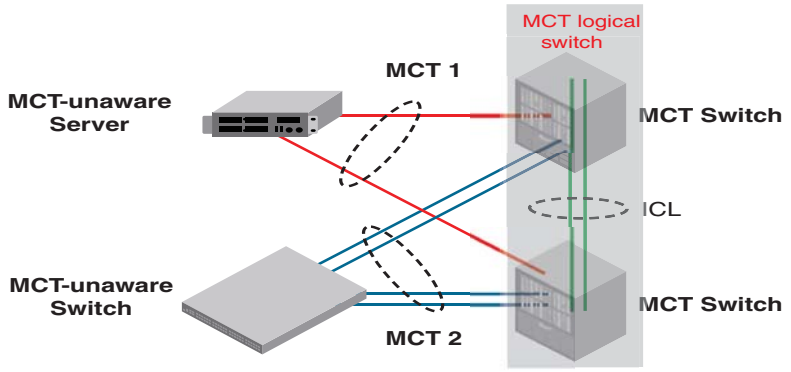
A Multi-Chassis Trunk (MCT) is a trunk that initiates at a single MCT-unaware server or switch and terminates at two MCT-aware switches.

Link Aggregation (LAG) trunks provide link level redundancy and increased capacity. However, LAG trunks do not provide switch level redundancy. If the switch to which the LAG trunk is attached fails, the entire LAG trunk loses network connectivity. With MCT, member links of the LAG are connected to two chassis. The MCT switches may be directly connected using an Inter-Chassis Link (ICL) to enable data flow and control messages between them.In this model, if one MCT switch fails, a data path will remain through the other switch.

In a MCT scenario, all links are active and can be load shared to increase bandwidth. In addition, traffic restoration can be achieved in milliseconds after an MCT link failure or MCT switch failure.

MCT is designed to increase network resilience and performance.

FIGURE 109 Chassis trunk example



Benefits to Multi-Chassis Trunking (MCT)

MCT provides the following benefits:

- Provides link level and switch level redundancy.
- Provides increased capacity because it utilizes all links (including redundant ones) for traffic transport. This contrasts with the use of the Spanning Tree Protocol, which does not use redundant links for transporting traffic.
- Provides traffic restoration in tens of milliseconds in case of link or switch failures.
- Allows servers and switches to have redundant connections to two switches and to fully utilize all links (including redundant ones) for traffic transport.
- Allows servers and switches to use standard link aggregation (802.3ad) to connect to redundant switches.
- MCT is easily deployed while enhancing existing multilayer switching without fundamentally changing the architecture.

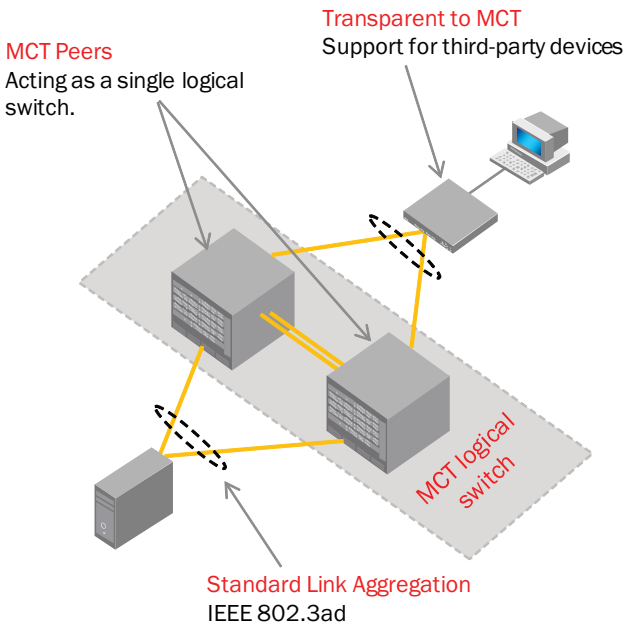
How Multi-Chassis Trunking works

The MCT is made up of the following:

- Sub-second failover in the event of a link, module, switch fabric, control plane, or node failure
- Layer 2 and Layer 3 forwarding (when using fast path forwarding) at the first hop regardless of VRRP-E state.
- Flow based load balancing rather than VLANs sharing across network links
- Ability to provide the resiliency regardless of the traffic type layer 3, layer 2 or non-IP (legacy protocols).
- Interaction with MRP to build larger resilient Layer 2 domains
- Enhancement to Link Aggregation Groups
- Provides nodal redundancy in addition to link and modular redundancy

- Operates at the physical level to provide sub-second failover

FIGURE 110 How Multi-Chassis Trunking works



MCT terminology

- ICL – Inter Chassis Link
- CCP – Cluster Communication Protocol
- CCEP – Cluster Client Edge Port
- CEP – Cluster Edge Port
- MCT VLANs - VLANs on which MCT clients are operating. These VLANs are explicitly configured in the MCT configuration by the user.
- RBridge ID – RBridge ID is a value assigned to MCT nodes and clients to uniquely identify them, and helps in associating Source MAC with a MCT node
- MDUP – MAC Database Update Protocol
- CL – Cluster Local MACs
- CCL – Cluster Client Local MACs
- CR – Cluster Remote MACs
- CCR – Cluster Client Remote MACs
- CCRR– Cluster Client RBridge Reachability
- MDB – Mac Data Base. The MDB can have multiple MAC entries for the same address
- FDB – Forwarding MAC Database. The FDM will have the best MAC only installed

Dynamic LAGs

MCT Client creates a single dynamic LAG towards the MCT nodes. For MCT nodes the dynamic Lag consists of two LAGs, each is configured on one of the MCT devices. A dynamic LAG runs Link Aggregation Control Protocol (LACP).

For the two dynamic LAGs of the MCT to behave as a single LAG from the MCT client's perspective, both of the dynamic LAGs should have the same LACP system ID and key, referred to as the MCT system ID and MCT key.

The MCT system ID and MCT key is uniquely defined for one MCT. They have the following attributes:

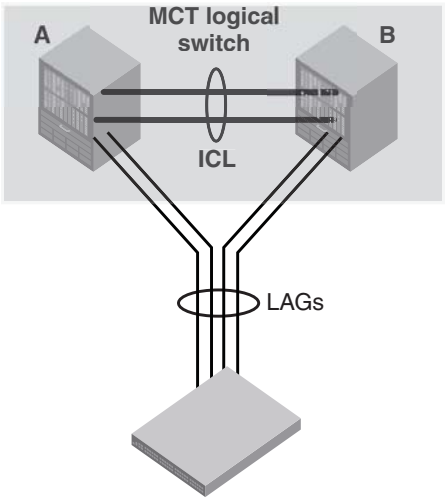
- MCT base system id = This is the system base MAC address (This is different for each system and is shown in the **show module** command output. The MAC displayed will be the first MAC address in the system).
- MCT system id = MCT base system id + cluster id
- MCT base key = 30000
- MCT key = MCT base key + client bridge id
- Each LAG link connects to one MCT node only

NOTE
Each MCT node has a unique cluster ID, and one MCT client ID.

MCT peers

Each MCT physical node, A and B, will act as an MCT peer and they are connected using an ICL. The pair of MCT peers will act as one logical switch for the access switch or server so that the MCT pair can connect using standard LAG to them. This is illustrated in Figure 111.

FIGURE 111 MCT peers



ICL traffic handling

An ICL link on the Dell device can be a single port or a static or LACP LAG. Non-MCT VLANs can co-exist with MCT VLANs on the ICL only on the NetIron MLX. For MCT VLANs, MAC learning is disabled on ICL ports.

L2 protocol packet handling

The default behavior is to forward or flood STP and RSTP BPDU packets.

If the **no cluster-l2protocol-forward** command is configured on global basis or **cluster-l2protocol-forward** disable is configured on a port, the STP protocol packets coming on the MCT VLANs are dropped.

All other L2 protocol packets will be flooded on the MCT VLANs or dropped. The **cluster-l2protocol-forward** command is not applicable to these protocol packets. It only applies to STP or RSTP BPDU packets.

Forwarding broadcast, multicast and unknown unicast traffic

Traffic received from non-ICL ports is forwarded the same way as non-MCT devices.Traffic received from an ICL port is not forwarded to the CCEP port if the peer MCT switch has the reachability to the same cluster client.

NOTE
When there is a double failure, the forwarding behavior will be unpredictable and there may be a complete traffic loss. For example, when both the ICL cluster link and any one leg of the client CCEP link fail. From the physical topology perspective, it may appear like a path is available, while traffic may not be forwarded.

Syncing interface MACs to peer MCT devices

The MCT device uses an interface MAC to identify the packets that are addressed to the switch. Such packets may be received by a peer MCT device. The peer MCT device switches packets over the ICL to the local MCT switch to be routed properly.

MCT L2 protocols

When configuring L2 protocols, you should consider the following.

MRP

- An ICL interface can not be configured as MRP secondary interface and vice-versa as the ICL cannot be BLOCKING.
- MRP can not be enabled on MCT CCEP port and vice-versa.

G.8032

- If the port is an ERP interface, it can not be enabled as a CCEP port.
- If the interface is ICL interface it can not be configured with MS, FS or RPL.
- G.8032 and MCT are not supported together.

STP

- The STP algorithm has been modified such that ICL never goes to blocking. ICL guard mechanism ensures that if ICL is going in blocking state then the port on which the superior BPDUs are being received is moved to BLOCKING state and ICL guard timer starts running on it. This timer runs as long as Superior BPDUs are received on this interface. As long as this timer runs on an interface the Superior BPDUs are dropped.
- The modified STP algorithm also ensures that the CCEP interface state on both the MCT peers is same.
- The CCEP STP state information between MCT peers is synchronized using messages that are sent over CCP.
- Only one of the MCT peers send BPDUs towards the MCT Client. It is decided by whosoever is the designated bridge on the ICL.
- New STP States:
  - BLK_BY_ICL state indicates that the superior BPDUs were being received on this interface which could have led to BLOCKING of ICL interface, due to which ICL port guard mechanism has been triggered on this port.
  - FWD_BY_MCT state indicates that the MCT peer has set the CCEP state to forwarding.
  - BLK_BY_MCT state indicates that the MCT peer has set the CCEP state to blocking.

MCT feature interaction

Use the following feature matrix when configuring MCT:

TABLE 102 MCT feature interaction matrix

Supported	Not Supported
LACP on both ICL and CCEP.	**MSTP, VSRP, RIP, OSPF, IS-IS, and BGP**
VRRP on the CCEP.	**ESI VLANs** on CCEP or ICL ports.
MRP and MRP II supported with the restriction that ICL port cannot be the secondary port of the MRP ring.	**802.1ad** on CCEP or ICL ports.
	GRE is not supported on the ICL ve interfaces
Flooding features (VLAN CPU protection, multicast flooding etc.) on cluster VLANs.	**DAI** on the CCEP ports.
UDLD as independent boxes.	**802.1ah** on CCEP or ICL ports.
Link OAM as independent boxes.	**VPLS** on CCEP or ICL ports.
802.1ag as independent boxes.	**VLL** on CCEP or ICL ports.
ARP as independent boxes.	**MPLS** on CCE or ICL ports.
STP and RSTP	**MSTP**

TABLE 102 MCT feature interaction matrix

Supported	Not Supported
L3 Routing - The IP address assignment is OK on CCEP ports for VRRP purpose. However, routing protocols would not be enabled on CCEP ports.	**Hitless Fail over** is NOT supported on the NetIron MLX, however it is compatible. If the operation is performed with cluster configuration the TCP session is reestablished. The MACs from the cluster peers will be revalidated and programmed accordingly. Dell recommends shutting down all the CCEP ports on the cluster node so that there is graceful failover and then hitless operation can be performed.
Port MAC Security on the node where it is programmed.	**Hitless Upgrade** is NOT supported, on the NetIron MLX, however it is compatible. If the operation is performed with cluster configuration the TCP session is reestablished. The MACs from the cluster peers will be revalidated and programmed accordingly. Dell recommends shutting down all the CCEP ports on the cluster node so that there is graceful failover and then hitless operation can be performed.
802.1x on the node where it is programmed.	**Multi-port MAC** are not supported on ICL or CCEP ports. Configuration will be rejected when trying to configure multi-port MAC addresses with a port mask which contains either a CCEP port or ICL port and vice-versa on the NetIron MLX.
Static MAC configuration - Static MACs are programmed on both local and remote peers as static entries.	

Configuration considerations

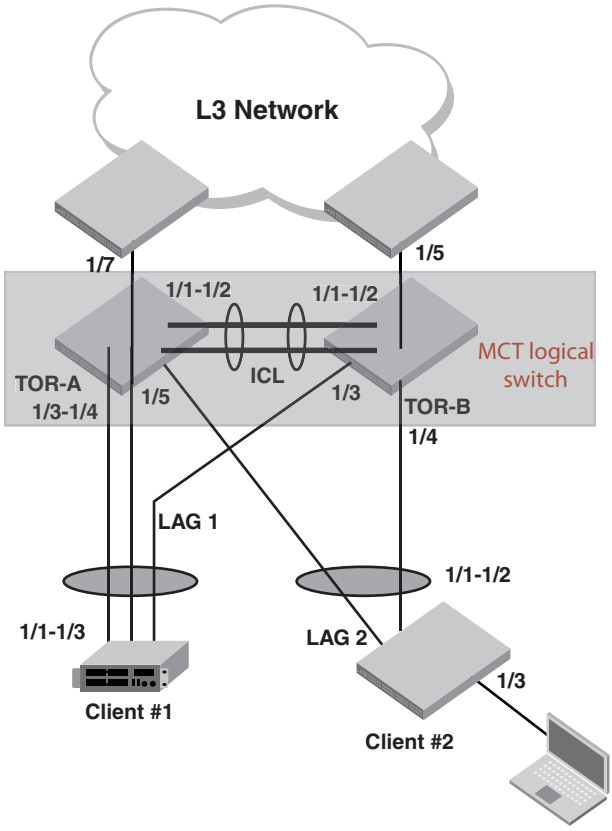
- ICL ports should be not be an untagged member of any VLAN. An ICL is preferably a LAG that provides port level redundancy and higher bandwidth for cluster communication.
- On the NetIron MLXe, ICL ports can be part of MCT VLANs as well as regular VLANs.
- MAC Database Update Protocol (MUDP) will synchronize all MAC entries for VLANs served by ICL link
- On Customer Client Edge Ports (CCEP), MCT does not support the following features: MPLS, VLL, VPLS, 802.1ah, 802.1ad and Routing protocols.
- Cluster ID should be same on both cluster switches
- Cluster rbridge ID should not conflict with any client rbridge id or the peer rbridge id
- Client rbridge ID is unique and it should be same on the cluster switches
- You can add to any ports to the session VLAN (For the purpose of adding a port to a LAG), but Dell recommends keeping only ICL ports during operation.
- An ICL interface cannot be configured as the CCEP port in any client
- CCEP ports cannot be configured as the ICL interface.
- CCEP ports on MCT node can be single port or LAG
- If ICL or client interfaces need to be configured as LAG interface then only the primary port of the LAG needs to be specified in the ICL or client configuration.
- Once the cluster is deployed, only the cluster member VLANs and client isolation mode can be modified. Other configurations are not allowed to change.

- Once the client is deployed, any configuration under client is not allowed to change.
- Clients can be added or deleted even if the cluster is deployed.
- When the cluster is undeployed all the clients in the cluster becomes inactive.
- As soon as a port is configured as an ICL port it is removed from default VLAN.
- If an ICL or CCEP is a LAG interface, the LAG has to be configured separately on each node.

Configuring MCT

Single level MCT example

FIGURE 112 Single level MCT



The following steps are task for configuring a MCT scenario as shown in Figure 112 on page 600.

NOTE
Save the current configuration files for both chassis operating in standalone mode before you begin creating a MCT.

TOR-A (Top of rack MCT capable switch)
See Figure 112 on page 600

Creating LAG-1

1. You can either assign a LAG **ID** explicitly or it will be automatically generated by the system. The LAG **ID** stays the same across system reload and hitless upgrade.

   The command to configure LAGs allows explicit configuration of the LAG ID for static and dynamic LAGs.

   To create a LAG with the LAG ID option, enter a command such as the following.

   ```
   NetIron(config)# lag 1 dynamic
   NetIron(config-lag-1)#
   ```

   Syntax: [no] lag <*name*> **[static | dynamic] [id** <*number*>**]**

   The **ID** parameter is optional. The value of the **ID** parameter that you can enter is from 1 to 256. If you do not enter a LAG **ID**, the system will generate one automatically. Once the LAG **ID** is generated the system will save it in the configuration file along with the LAG name, therefore the value will stay the same across system reload.

   NOTE
   The LAG id parameter is for static and dynamic LAGs only. No explicit configuration of a LAG id is allowed on keepalive LAGs.

   The **static** parameter specifies that the LAG with the name specified by the <*lag-name*> variable will be configured as a static LAG.

   The **dynamic** option specifies that the LAG with the name specified by the <*lag-name*> variable will be configured as a dynamic LAG.

2. Define the ports the LAG will be using as shown in the following.

   ```
   NetIron(config-lag-1)# ports ethernet 1/1 to 1/2
   ```

   Syntax: [no] ports ethernet [slot/port] | to |[slot/port]

   Use the appropriate **[slot/port]** variable to specify a Ethernet port within the LAG that you want to enable.

3. The primary port must be explicitly assigned using the **primary-port** command. To designate the primary port for the static LAG “1”, use the following command.

   ```
   NetIron(config-lag-1)# primary-port 1/1
   ```

   Syntax: [no] primary-port<*slot/port*>

   Once a primary port has been configured for a LAG, all configurations that apply to the primary port are applied to the other ports in the LAG.

   NOTE
   This configuration is only applicable for configuration of a static or dynamic LAGs

4. After configuring a LAG, you must explicitly enable it before it begins aggregating traffic. This task is accomplished by executing the deploy command within the LAG configuration. After the deploy command runs, the LAG is in the aggregating mode. Only the primary port within the LAG is available at the individual interface level. Any configuration performed on the primary port applies to all ports within the LAG. The running configuration will no longer display deployed LAG ports other than the primary port.

To deploy a LAG, at least one port must be in the LAG and the primary port must be specified for non keep-alive LAGs. After a non keep-alive LAG is deployed, a trunk is formed. If there is only one port in the LAG, a single port is formed. For a dynamic LAG, LACP is stared for each LAG port.

Use a command such as the following to deploy LAG 1

```
NetIron(config-lag-1)# deploy
```

Syntax: [no] deploy [forced | passive]

When the **deploy** command is executed:

- For a static and dynamic LAGs, the current veto mechanism is invoked to make sure the LAG can be formed. If the LAG is not vetoed, a **no** is formed with all the ports in the LAG.
- For dynamic LAGs, LACP is activated on all LAG ports. When activating LACP, use active mode if passive is not specified; otherwise, use passive mode.
- For a keep-alive LAGs, a LAG is formed, and LACP is started on the LAG port.

Once the deploy command is issued, all LAG ports will behave like a single port.

If the **no** deploy command is executed, then the LAG is removed. For dynamic LAGs, LACP is de-activated on all of the LAG ports.

If the **no deploy** command is issued and more than 1 LAG port is not disabled the command is aborted and the following error message is displayed: “Error 2 or more ports in the LAG are not disabled, un-deploy this LAG may form a loop - aborted.” Using the **forced** keyword with the **no deploy** command in the previous situation, the un-deployment of the LAG is executed.

5. Assign a name to an individual port within a LAG using the **port-name** command within the LAG configuration as shown in the following.

```
NetIron(config-lag-1)# port-name ICL-to-TOR-B:1/1 ethernet 1/1
NetIron(config-lag-1)# port-name ICL-to-TOR-B:1/2 ethernet 1/2
```

Syntax: [no] port-name <*text*> **ethernet [slot/port] | pos [slot/port]**

The <*text*> variable specifies the port name. The name can be up to 50 characters long.

Use the **ethernet** option with the appropriate **[slot/port]** variable to apply the specified name to an Ethernet port within the LAG.

Use the **pos** option with the appropriate **[slot/port]** variable to apply the specified name to a Packet-over-SONET port within the LAG.

Creating LAG 2

See Figure 112 on page 600 and “Creating LAG-1” on page 601 for additional information on creating a LAG.

1. Create LAG 2 as shown below.

```
NetIron(config)# lag 2 dynamic id 2
NetIron(config-lag-2)#
```

2. Define the ports the LAG will be using.

```
NetIron(config-lag-2)# ports ethernet 1/3  to 1/4
```

3. Deploy the LAG 2 as shown below.

```
NetIron(config-lag-2)# deploy
```

4. Assign a name to an individual port within a LAG.

```
NetIron(config-lag-2)# port-name lag-client-1:1/1 ethernet 1/3
NetIron(config-lag-2)# port-name lag-client-1:1/2 ethernet 1/4
```

Creating LAG 3

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. Create LAG 3 as shown below.

```
NetIron(config)# lag 3 dynamic id 3
NetIron(config-lag-3)#
```

2. Define the ports the LAG will be using.

```
NetIron(config-lag-3)# ports ethernet 1/5
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
NetIron(config-lag-3)# primary-port 1/5
```

4. Deploy the LAG 3 as shown below.

```
NetIron(config-lag-3)# deploy
```

5. Assign a name to an individual port within a LAG.

```
NetIron(config-lag-2)# port-name lag-client-2:1/1 ethernet 1/5
NetIron(config-lag-2)# port-name lag-client-1:1/2 ethernet 1/4
```

Enable layer 2 switching

1. By default, PowerConnect devices supports routing over layer 2 switching. You can enable layer 2 switching globally or on individual port using the **no route-only** command. The **no route-only** and **route-only** commands prompts you for whether or not you want to change the "route-only" behavior. You must enter **y** if you want to proceed or **n** if you do not. To enable Layer 2 switching only on a specific interface, go to the interface configuration level for that interface, and add the **no route-only** command.

   Use commands such as the following to enable Layer 2 switching.

```
NetIron(config)# no route-only
```

   Syntax: [no] route-only

Creating MCT VLANs

See Figure 112 on page 600 and "MCT VLAN1" on page 603 for additional information on creating VLANs.

MCT VLAN1

1. At the global CONFIG level assign an ID to the VLAN.

```
NetIron(config)# vlan 2
```

2. Add ports to the VLAN and specify if the ports are tagged or untagged.

```
NetIron(config-vlan-2)# tagged e 1/1 to 1/8
NetIron(config-vlan-2)# no untag ether 1/1 to 1/2
```

VLAN 2

See Figure 112 on page 600 and "MCT VLAN1" on page 603 for additional information on creating VLANs.

1. Configure the VLAN name for VLAN 2.

```
NetIron(config)# vlan 2 name client-vlan
```

2. Add ports to the VLAN and specify if the ports are tagged or untagged.

```
NetIron(config-vlan-2)# untag ether 1/3 to 1/7
NetIron(config-vlan-2)# tagged ether 1/1 to 1/2
```

Create the session VLAN

See Figure 112 on page 600 and "MCT VLAN1" on page 603 for additional information on creating VLANs.

1. At the global CONFIG level assign an ID to the VLAN .

```
NetIron(config)# vlan 4090 name Session-VLAN
```

2. Add ports to the VLAN and specify if the ports are tagged or untagged.

```
NetIron(config-vlan-4090)# tagged ether 1/1 to 1/2
```

3. Configure a virtual routing interface on each IP protocol VLAN, then configure the appropriate IP routing parameters on each of the virtual routing interfaces.

```
NetIronNetIron(config-vlan-4090)# tagged ether 1/1 to 1/2
NetIron(config-vlan-4090)# router-interface ve 100
```

Assign the hostname

To configure a system name, enter commands such as the following.

```
NetIron(config)# hostname TOR-A
```

Enabling interfaces

The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the ports on a device for ethernet interfaces 1/1, 1/3, and 1/5, enter the following commands.

```
NetIron(config)# interface ether 1/1
NetIron(config-if-e10000-1/1)# enable

NetIron(config)# interface ether 1/3
NetIron(config-if-e10000-1/3)# enable

NetIron(config)# interface ether 1/5
NetIron(config-if-e10000-1/5)# enable
```

Syntax: [no] enable

Assigning a port name

A port name can be assigned to help identify interfaces on the network. You can assign a port name to physical ports, virtual routing interfaces, and loopback interfaces. To assign a name to a ports 1/6 and 1/7, enter the following commands.

```
NetIron(config)# interface ethe 1/6
NetIron(config-if-e10000-1/6)# port-name CEP-PC
NetIron(config-if-e10000-1/6)# enable
```

```
NetIron(config)# interface ethe 1/7
NetIron(config-if-e10000-1/7)# port-name to-L3-ECMP
NetIron(config-if-e10000-1/7)# enable
```

Syntax: [no] port-name <text>

Syntax: [no] enable

The <text> parameter is an alphanumeric string. The name can have up to 255 characters on a device and can include blanks. You do not need to use quotation marks around the string, even when it contains blanks.

Adding a virtual interface

Add a virtual interface and configure an IP address on the interface by entering commands such as the following.

```
NetIron(config)# interface ve 100
NetIron(config-vif-100)# ip address 1.1.1.1/24
```

Syntax: [no] interface [ve <*ve-id*>]

Syntax: [no] ip address <*ip-addr*> <*ip-mask*>

The <*ve-id*> variable allows you to specify a VE interface ID.

Configuring the cluster operation mode

See Figure 112 on page 600

1. To configure a device with cluster id 1, enter a command such as the following.

   ```
   NetIron(config)# cluster TOR 1
   ```

   Syntax: [**no**] **cluster** <cluster-name> <cluster-id>

   The <cluster-name> parameters specify the cluster name with a limit of 64 characters.

   The <cluster-id> parameters specify the cluster ID (1-65535).

2. Configure the local rbridge id for the cluster. This rbridge id is used by the peer to communicate with this cluster node. To configure the local rbridge, enter a command such as following

   ```
   NetIron(config-cluster-TOR)#rbridge-id 1
   ```

   Syntax: [no] rbridge-id <id>

   The <id> parameters specify the remote bridge id. Possible values are 1 - 35535 (16 bit value).

3. The cluster session VLAN can be in the range 1-4090, but cannot be default VLAN. A check is made during the cluster deploy in addition to a dynamic check. The default VLAN cannot be changed to a VLAN which is already defined as cluster session. Enter a command such as the following to create the session VLAN.

   ```
   NetIron(config-cluster-TOR)# session-vlan 4090
   ```

   Syntax: [no] session-vlan <vlan-id>

   The <vlan-id> parameters specify the VLAN range. Possible values are 1 - 4090.

4. Specify the VLAN range on which cluster is operating. This would be the range for which there would be MAC synchronization. Multiple VLAN ranges would be supported for the configuration. Enter a command such as the following to create the member VLAN.

```
NetIron(config-cluster-TOR)# member-vlan 2
```

Syntax: [no] member-vlan <x> [to <y>]

NOTE
The VLAN range is allowed to change even if cluster is deployed.

5. Specify the ICL for the cluster.The ICL interface can be a single link or trunk port. If it is a trunk port, it should be the primary port of the trunk . Only one ICL is supported. Enter a command such as the following to create the ICL for the cluster.

```
NetIron(config-cluster-TOR)#icl TOR ethernet 1/1
```

Syntax: [no] icl <icl-name> **ethernet** x/y

The <icl-name> parameter can be up to 64 characters in length.

The **ethernet** x/y parameter is the ICL interface.

6. Specify the rbridge and ICL for the peers by entering a command such as the following.

```
NetIron(config-cluster-TOR)# peer 1.1.1.2 rbridge-id 2 icl TOR
```

Syntax: [no] peer <peer-ip> **rbridge-id** <peer-rbridge> **icl** <map-icl>

The< peer-ip >parameter should be in same subnet as that of cluster management interface.

The <peer-rbridge> parameter should be different from cluster rbridge and any other client in the cluster

The <map-icl> parameter is the ICL name to reach this cluster peer.

7. The cluster can be deployed separately without any clients configured. The **deploy** command brings the cluster into effect. The following can be changed when the cluster is deployed:
   - Client isolation mode
   - Member VLANs
   - Clients added and removed.
8. The **deploy** command also preforms a consistency check of the entire cluster configuration. If anything is amiss, an error message is sent.The following specific information is checked during deployment:
   - If the cluster management VLAN is configured
   - If the cluster peer is configured
   - If the cluster ICL is configured

   Enter a command such as the following to deploy the cluster configuration.

```
NetIron(config-cluster-TOR)# deploy
```

Syntax: [no] deploy

Creating cluster client 1

See Figure 112 on page 600

1. Create a cluster client instance and change the mode to the client instance. If an instance is already present, then directly change the mode to the client instance mode.

```
NetIron(config-cluster-TOR)# client client-1
```

Syntax: [no] client <client-name>

The <client-name> parameter can be 64 characters (maximum).

2. Configure the local RBridge IDfor the cluster. This RBridge ID is used by the peer to communicate with this cluster node. To configure the local rbridge, enter a command such as following

```
NetIron(config-cluster-TOR-client-1)#rbridge-id 100
```

Syntax: [no] rbridge-id <id>

The <id> parameters specify the local bridge id. Possible values are 1 - 35535 (16 bit value).

3. The cluster session VLAN is the VLAN used by the cluster for control operations. CCP protocol runs over this VLAN. The interface can be a single link or LAG port. If it is LAG port, it should be the primary port of the LAG.

```
NetIron(config-cluster-TOR-client-1)#client-interface ether 1/3
```

Syntax: [no] client-interface <interface> interface : ethernet x/y

The **ethernet** x/y parameter is the ethernet interface.

4. Deploy the cluster client. If cluster is not deployed, the configuration will be taken but the client state machine will not be started. The consistency checks for client will be done at the time of client deploy. The following configuration checks will be preformed:

- Client interface is configured
- Client interface is not same as any other client interface or ICL interface
- Client RBridge ID is not same as cluster rbridge or any peer rbridge

Once the client is deployed, the configuration inside the client will not be allowed to change, To deploy the client configuration, enter a command such as the following.

```
NetIron(config-cluster-TOR-client-1)deploy
```

Syntax: [no] deploy

Create cluster client 2

See Figure 112 on page 600 and "Create cluster client 1" on page 613 for additional information for creating cluster clients.

1. Create a cluster client instance.

```
NetIron(config-cluster-TOR)# client client-2
```

2. Configure the client RBridge ID.

```
NetIron(config-cluster-TOR-client-2)#rbridge-id 200
```

3. Create a cluster client interface.

```
NetIron(config-cluster-TOR-client-2)#client-interface ether 1/5
```

4. Deploy the cluster client.

```
NetIron(config-cluster-TOR-client-2)deploy
```

TOR-B

Creating LAG-1

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. Create a LAG with the LAG ID option.

```
NetIron(config)# lag 1 dynamic id 1
NetIron(config-lag-1)#
```

2. Define the port the LAG will be using.

```
NetIron(config-lag-1)# ports ethernet 1/6
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
NetIron(config-lag-1)# primary-port 1/1
```

4. Deploy a LAG as shown below.

```
NetIron(config-lag-1)# deploy
```

Assign a name to an individual port within a LAG.

```
NetIron(config-lag-1)# port-name lag-client-2:1/2 ethernet 1/6
```

Creating LAG 2

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
NetIron(config)# lag 2 dynamic id 2
NetIron(config-lag-2)#
```

2. Define the port the LAG will be using.

```
NetIron(config-lag-2)# ports ethernet 1/7
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
NetIron(config-lag-2)# primary-port 1/7
```

4. Deploy a LAG as shown below.

```
NetIron(config-lag-2)# deploy
```

5. Assign a name to an individual port within a LAG.

```
NetIron(config-lag-2)# port-name lag-client-3:1/2 ethernet 1/7
```

Creating LAG 3

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
NetIron(config)# lag 3 dynamic id 3
NetIron(config-lag-3)#
```

2. Define the port the LAG will be using.

```
NetIron(config-lag-3)# ports ethernet 1/3 to 1/4
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
NetIron(config-lag-3)# primary-port 1/3
```

4. Deploy a LAG as shown below.

```
NetIron(config-lag-3)# deploy
```

5. Assign a name to an individual port within a LAG.

```
NetIron(config-lag-3)# ICL-to-TOR-A:1/3 ethernet 1/3
NetIron(config-lag-3)# ICL-to-TOR-A:1/4 ethernet 1/4
```

Creating LAG 4

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
NetIron(config)# lag 4 dynamic id 4
NetIron(config-lag-4)#
```

2. Define the port the LAG will be using.

```
NetIron(config-lag-4)# ports ethernet 1/5
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
NetIron(config-lag-4)# primary-port 1/5
```

4. Deploy a LAG as shown below.

```
NetIron(config-lag-4)# deploy
```

5. Assign a name to an individual port within a LAG.

```
NetIron(config-lag-3)# lag-client-1:1/2 ethernet 1/5
```

Enable layer 2 switching

1. By default, PowerConnect devices supports routing over layer 2 switching. You can enable layer 2 switching globally or on individual port using the **no route-only** command. The **no route-only** and **route-only** commands prompts you for whether or not you want to change the "route-only" behavior. You must enter **y** if you want to proceed or **n** if you do not. To enable Layer 2 switching only on a specific interface, go to the interface configuration level for that interface, and add the **no route-only** command.

   Use the following command to enable Layer 2 switching.

```
NetIron(config)# no route-only
```

   Syntax: [no] route-only

Creating VLANs

See Figure 112 on page 600

VLAN 1

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN.

```
NetIron(config)# vlan 2
```

2. Add ports to that VLAN and specify if the ports are tagged or untagged.

```
NetIron(config-vlan-2)# tagged e 1/1 to 1/8
NetIron(config-vlan-2)# no untag ether 1/3 to 1/4
```

VLAN 2

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN 2.

```
NetIron(config)# vlan 2 client-vlan
```

2. Add ports to that VLAN and specify if the ports are tagged or untagged.

```
NetIron(config-vlan-2)# untag ether 1/5 to 1/7
NetIron(config-vlan-2)# tagged ether 1/3 to 1/4
```

Create the session VLAN

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN 4090.

```
NetIron(config)# vlan 4090 name Session-VLAN
```

Syntax: [no] vlan <*vlan-id*>name <*vlan-name*>

VLAN IDs can be in the range of 1 – 4090. Use the **no** form of the command to delete the VLAN from the configuration.

The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

In addition to a VLAN number, you can assign a name to a VLAN by entering name <*vlan-name*>. Enter up to 32 characters for name.

2. Add ports to the VLAN and specify if the ports are tagged or untagged.

```
NetIron(config-vlan-4090)# tagged ether 1/3 to 1/4
```

3. Configure the appropriate IP routing parameters on each of the virtual routing interfaces.

```
NetIron(config-vlan-4090)# tagged ether 1/3 to 1/4
NetIron(config-vlan-4090)# router-interface ve 100
```

Assign the hostname

Configure a system name for the device and save the information locally in the configuration file for future reference. The information is not required for system operation but recommended. When you configure a system name, it replaces the default system name in the CLI command prompt.

To configure a system name, enter a command such as the following.

```
NetIron(config)# hostname TOR-B
```

Syntax: [no] hostname <*string*>

The name can be up to 255 alphanumeric characters. The text strings can contain blanks.

Enabling interfaces

The ports can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the ports on a device for ethernet interfaces 1/3, 1/5, 1/6,and 1/7, enter the following commands.

```
NetIron(config)# interface ether 1/3
NetIron(config-if-e10000-1/3)# enable

NetIron(config)# interface ether 1/5
NetIron(config-if-e10000-1/5)# enable

NetIron(config)# interface ether 1/6
NetIron(config-if-e10000-1/6)# enable

NetIron(config)# interface ether 1/7
NetIron(config-if-e10000-1/7)# enable
```

Syntax: [no] enable

Adding a virtual interface

Add a virtual interface and configure an IP address on the interface by entering commands such as the following.

```
NetIron(config)# interface ve 100
NetIron(config-vif-100)# ip address 1.1.1.2/24
```

Syntax: [no] interface [ve <*ve-id*>]

Syntax: [no] ip address <*ip-addr*> <*ip-mask*>

The <*ve-id*> variable allows you to specify a VE interface ID.

Configuring the cluster operation mode

The cluster can be deployed separately without any client configured. When the cluster is deployed, it will check all the deployed clients and start the state machine for the clients. See Figure 112 on page 600.

1. Configure one cluster ID or name on the device so that all route-reflector clients for the device become members of the cluster. To configure a device with cluster id 1, enter the following command.

   ```
   NetIron(config)# cluster TOR 1
   ```

   Syntax: [**no**] **cluster** <cluster-name> <cluster-id>

   The <cluster-name> parameters specify the cluster name with a limit of 64 characters.

   The <cluster-id> parameters specify the cluster ID (1-65535). The default is the device ID.

2. Configure the remote bridge id cluster on the device so all clients for the device become members of the cluster.

```
NetIron(config-cluster-TOR)#rbridge-id 2
```

Syntax: [no] rbridge-id <id>

The <id> parameters specify the remote bridge id. Possible values are 1 - 35535 (16 bit value).

3. The cluster session VLAN is in range 1-4090 but cannot be default VLAN. A check is made during the cluster deploy and in addition to a dynamic check. The default VLAN cannot be changed to a VLAN which is already defined as cluster session

```
NetIron(config-cluster-TOR)# session-vlan 4090
```

Syntax: [no] session-vlan <vlan-d>

The <vlan-id> parameters specify the VLAN range. Possible values are 1 - 4090.

4. Specify the VLAN range on which cluster is operating. This would be the range for which there would be MAC synchronization. Multiple VLAN ranges would be supported for the configuration. Enter a command such as the following to create the member VLAN.

```
NetIron(config-cluster-TOR)# member-vlan 2
```

Syntax: [no] member-vlan <x> to <y>

NOTE
The VLAN range is allowed to change even if cluster is deployed.

The new VLAN range will over-ride the previous configured range.

5. Specify the ICL for the cluster.The ICL interface can be a single link or trunk port. If it is a trunk port, it should be the primary port of the trunk. Only one ICL is supported.

```
NetIron(config-cluster-TOR)#icl TOR ethernet 1/3
```

Syntax: [no] icl <icl-name> **ethernet** x/y

The <icl-name> parameter can be 64 characters (maximum).

The **ethernet** x/y parameter is the ICL interface.

6. Specify the rbridge and ICL for the peers by entering a command such as the following.

```
NetIron(config-cluster-TOR)# peer 1.1.1.1 rbridge-id 1 icl TOR
```

Syntax: [no] peer <peer-ip> **rbridge-id** <peer-rbridge> **icl** <map-icl>

The< peer-ip >parameter should be in same subnet as that of cluster management interface.

The <peer-rbridge> parameter should be different from cluster rbridge and any other client in the cluster

The <map-icl> parameter is the ICL name to reach this cluster peer.

7. Clusters can be deployed separately without any client configured. The **deploy** command brings the cluster into effect. Once the cluster is deployed, the configuration inside the cluster can not be changed. The **deploy** command also preforms a consistency check of the entire cluster configuration. If anything is amiss, an error message is sent.The specific information checked during deploy:

    - If the cluster management VLAN is configured
    - If the cluster peer is configured
    - If the cluster ICL is configured

Enter a command such as the following to deploy the cluster configuration.

```
NetIron(config-cluster-TOR)# deploy
```

Syntax: [no] deploy

Create cluster client 1

1. Create a cluster client instance and change the mode to the client instance. If an instance is already present, then directly change the mode to client instance mode.

   ```
   NetIron(config-cluster-TOR)# client client-1
   ```

   Syntax: [no] client <client-name>

   The <client-name> parameter can be 64 characters (maximum).

2. Configure the remote bridge id cluster on the device so all clients for the device become members of the cluster.

   ```
   NetIron(config-cluster-TOR-client-1)#rbridge-id 100
   ```

   Syntax: [no] rbridge-id <id>

   The <id> parameters specify the remote bridge id. Possible values are 1 - 35535 (16 bit value).

3. Create a cluster client interface. The interface can be a single link or trunk port. If it is trunk port, it should be the primary port of the trunk.

   ```
   NetIron(config-cluster-TOR-client-1)#client-interface ether 1/5
   ```

   Syntax: [no] client-interface <interface> interface : ethernet x/y

   The **ethernet** x/y parameter is the ethernet interface.

4. Deploy the cluster client. If cluster is not deployed, the configuration will be taken but the client FSM will not be started. The consistency checks for client will be done at the time of client deploy. The following configuration checks will be preformed:

   - Client interface is configured
   - Client interface is not same as any other client interface or ICL interface
   - Client RBridge ID is not same as cluster rbridge or any peer rbridge

   Once the client is deployed, the configuration inside the client will not be allowed to change, To deploy the client configuration, enter a command such as the following.

   ```
   NetIron(config-cluster-TOR-client-1)deploy
   ```

   Syntax: [no] deploy

Create cluster client 2

See Figure 112 on page 600 and "Create cluster client 1" on page 613 for additional information on creating cluster clients.

1. Create a cluster client instance and change the mode to the client instance mode.

   ```
   NetIron(config-cluster-TOR)# client client-2
   ```

2. Configure the remote bridge id cluster on the device so all clients for the device become members of the cluster.

   ```
   NetIron(config-cluster-TOR-client-2)#rbridge-id 200
   ```

3. Create a cluster client interface.

```
NetIron(config-cluster-TOR-client-2)#client-interface ether 1/6
```

4. Deploy the cluster client.

```
NetIron(config-cluster-TOR-client-2)deploy
```

Create cluster client 3

See Figure 112 on page 600 and "Create cluster client 1" on page 613 for additional information on creating cluster clients.

1. Create a cluster client instance and change the mode to the client instance mode.

```
NetIron(config-cluster-TOR)# client client-3
```

2. Configure the remote bridge id cluster on the device so all clients for the device become members of the cluster.

```
NetIron(config-cluster-TOR-client-3)#rbridge-id 300
```

3. Create a cluster client interface. The interface can be a single link or LAG port. If it is LAG port, it should be the primary port of the LAG.

```
NetIron(config-cluster-TOR-client-2)#client-interface ether 1/7
```

4. Deploy the cluster client.

```
NetIron(config-cluster-TOR-client-2)deploy
```

Configuring Client-1

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. Create LAG 1 as shown below.

```
NetIron(config)# lag 1 dynamic id 1
NetIron(config-lag-1)#
```

2. Define the ports the LAG will be using.

```
NetIron(config-lag-1)# enable ethernet 1/1 to 1/3
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
NetIron(config-lag-1)# primary-port 1/1
```

4. Deploy the LAG as shown below.

```
NetIron(config-lag-1)# deploy
```

5. Assign a name to an individual port within a LAG.

```
NetIron(config-lag-1)# port-name lag-to-TOR-A ethernet 1/1
NetIron(config-lag-1)# port-name lag-to-TOR-B ethernet 1/3
```

6. The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the port on a device for ethernet interface 1/1, enter the following command.

```
NetIron(config-if-e10000-1/1)# enable
```

Syntax: [no] enable

Configuring Client 2

See Figure 112 on page 600 and "Creating LAG-1" on page 601 for additional information on creating a LAG.

1. Create a LAG with the LAG ID option, enter a command such as the following.

```
NetIron(config)# lag 1 dynamic id 1
NetIron(config-lag-1)#
```

2. Define the ports the LAG will be using.

```
NetIron(config-lag-1)# ports ethernet 1/1 to 1/2
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
NetIron(config-lag-1)# primary-port 1/1
```

4. Deploy the LAG as shown below.

```
NetIron(config-lag-1)# deploy
```

5. Assign a name to an individual port within a LAG.

```
NetIron(config-lag-1)# port-name lag-to-TOR-A ethernet 1/1
NetIron(config-lag-1)# port-name lag-to-TOR-B ethernet 1/2
```

6. The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the port on a device for ethernet interface 1/1 and 1/3, enter the following commands.

```
NetIron(config)# interface ether 1/1
NetIron(config-if-e10000-1/1)# enable

NetIron(config)# interface ether 1/3
NetIron(config-if-e10000-1/3)# enable
```

   Syntax: [no] enable

7. Assign a name to an individual port within a LAG.

```
NetIron(config-if-e10000-1/3)# port-name host-to-PC
NetIron(config-if-e10000-1/3)# enable
```

Optional cluster operation features

A cluster can operate in two modes:

Cluster Failover Mode

Fast-failover (default) - As soon as the ICL interface goes down the CCP goes down. All the remote MACs are flushed.

Slow-failover - Even if the ICL interface goes down the CCP waits for the hold-time before making the CCP down. Remote MACs are flushed only when the CCP is down.

To disable the fast-failover mode, enter a command such as the following.

```
NetIron(config-cluster-TOR)#peer 1.1.1.1 disable-fast-failover
```

Syntax: [no] peer <peer-ip> **disable-fast-failover**

Client isolation mode

NOTE
The CLI will allow modification of the **client-islolation** mode on MCT cluster nodes even when the cluster is deployed. You must create the same isolation mode on both cluster nodes.

Clients can operate in two modes:

Loose mode (**default**): When the CCP goes down the client performs the Master/Slave negotiation. After negotiation, the Slave shuts down its client ports whereas the Master client ports continue to forward the traffic (keep-alive VLAN configured).

If the keep-alive VLAN is not configured, both nodes become master.

```
NetIron(config-cluster-TOR)#client-isolation loose
```

Strict mode: When the CCP goes down, the client interfaces on both the cluster nodes are administratively shutdown. In this mode, the client is completely isolated from the network if CCP is not operational.

```
NetIron(config-cluster-TOR)#client-isolation strict
```

Syntax: [no] client-isolation strict

Shutdown all client interfaces

Use the **client-interfaces shutdown** command when preforming a hitless-upgrade operation. This command can be used to shutdown all the local client interfaces in the cluster. This would result in failover of traffic to the cluster peer.

```
NetIron(config-cluster-TOR)#client-interfaces shutdown
```

Syntax: [no] client-interfaces shutdown

Displaying cluster information

Use the **show cluster** command to display the entire cluster configuration and **show tech cluster** commmand to dsplay cluster configuration and operation information. See the *PowerConnect B-MLXe Diagnostic Reference* for additional information on these commands.

Keep-alive VLAN

CCRR message are used to exchange information between peers. When the CCP is up, CCRR messages are sent over CCP. When the CCP client reachability is down, you can use the **keep-alive-vlan** command under cluster context so CCRR messages are periodically sent over the keep-alive-vlan. Only one VLAN can be configured as a **keep-alive-vlan**. The VLAN can be any VLAN configured in the system

```
NetIron(config-cluster-TOR)#keep-alive-vlan 10
```

Syntax: [no] keep-alive-vlan <vlan-id>

The <vlan_id> parameters specify the VLAN range. Possible values are 1 - 4090

When the CCP is down:

- If **keep-alive-vlan** is configured, then CCRR messages are sent periodically for every 1 second over that VLAN.
- When CCP is down and keep-alive vlan is configured, Master/Slave selection is based on following criteria.
    1. If one node's CCEPs are up and other node's CCEPs are down then the node with local CCEPs down becomes Slave
    2. Otherwise, the node with higher RBridge ID becomes Slave.
- If no packetsare received from the peer for a period of 3 seconds, then the peer box is considered down.
- If **keep-alive-vlan** is not configured and both the peers are up, then both peers keep forwarding the traffic independently.

Keep-alive timers and hold-time

To specify the **keep-alive timers** and **hold-time** for the peers, enter a command such as the following.

```
NetIron(config-cluster-TOR)# peer 1.1.1.1 timers keep-alive 40 hold-time 120
```

Syntax: [no] peer <peer-ip> **timers keep-alive** <keep-alive time> **hold-time** <hold-time>

The <peer-ip> parameter should be in same subnet as that of cluster management interface.

The <keep-alive time> parameter can be 0 to 21845 (default 30 seconds.)

The <hold-time> parameter can be 3 to 65535 (default 90 seconds) must be at least 3 times the keep alive time.

NOTE
Keep-alive-vlan and keep-alive timers are not related. The keep-alive timer is used by CCP.

L2 protocol forwarding

MCT will forward or drop L2 protocol packets when corresponding features are disabled. The packets will either be forwarded as regular multicast that floods to the VLAN or dropped. When forwarded, the packet received from ICL will not be forwarded to CCEP port if the peer MCT switch has the reachability to the same cluster client.

When designing a network, the ICL port or LAG must have enough bandwidth to support all the traffic from clients (in case of client links connected to one node failure case).

For L2 forwarding, appropriate CAM profiles need to be used to be able to program all the MAC entries into the CAM (especially when using LAG interfaces on MCT nodes for ICL and client interfaces).

By default , MCT acts as a hub for STP, or RSTP. Switches connected to MCT can run STP normally. When STP, RSTP, or MSTP is enabled, the L2 protocol forwarding configuration is ignored and has no effect.

To configure L2 protocol forwarding globally, enter a command such as the following.

```
NetIron(config)#cluster-l2protocol-forward
```

To disable L2 protocol forwarding on an interface, enter a command such as the following.

```
NetIron(config-if-e1000-1/2)#cluster-l2protocol-foward disable
```

To remove L2 protocol forwarding configuration on an interface, enter a command such as the following

```
NetIron(config-if-e1000-1/2)#no cluster-l2protocol-foward <enable | disable>
```

Syntax: [no]cluster-l2protocol-forward [enable | disable]

Interface level configuration overwrites the global level configuration.

TABLE 103 L2 protocol forwarding action –MCT switch and non - MCT switch

Protocol	Destination MAC	Non- MCT switch forwarding action	MCT switch forwarding action
Untagged 802.1Q BPDU	01-80-c2-00-00-00	Flood to the VLAN	Forward to ports that are enabled by the **cluster-l2protocol-forward** command.
Tagged 802.1Q BPDU	01-80-c2-00-00-00	Flood to the VLAN	Forward to ports that are enabled by the **cluster-l2protocol-forward** command.
802.1Q Provider BPDU	01-80-c2-00-00-08		Same as non-MCT switch
802.3 Slow Protocols (e.g. LACP)	01-80-c2-00-00-02	Dropped	Same as non-MCT switch
802.1X PAE address	01-80-c2-00-00-03	Dropped	Same as non-MCT switch
802.1Q Provider Bridge GVRP	01-80-c2-00-00-0D	Flood to the VLAN	Same as non-MCT switch
802.1AB LLDP	01-80-c2-00-00-0E	Flood to the VLAN	Same as non-MCT switch
802.1D GMRP	01-80-c2-00-00-20	Flood to the VLAN	Same as non-MCT switch
802.1Q GVRP	01-80-c2-00-00-21	Flood to the VLAN	Same as non-MCT switch
Foundry MRP (Metro Ring Protocol)	03-04-80-00-00-00	Flood to the VLAN	Same as non-MCT switch
Foundry FDP (Foundry Discovery Protocol)	01-e0-52-cc-cc-cc	Flood to the VLAN	Same as non-MCT switch
CDP	01-00-00-cc-cc-cc	Flood to the VLAN	Same as non-MCT switch
PVST	01-00-0c-cc-cc-cd	Flood to the VLAN	Same as non-MCT switch
SuperSpan	03-80-c2-xx-xx-00	Flood to the VLAN	Same as non-MCT switch
VSRP Control	03-04-80-00-01-00	Flood to the VLAN	Same as non-MCT switch
VSRP Source	03-04-80-00-01-01	Flood to the VLAN	Same as non-MCT switch
Loop detection MAC	Base MAC address | 0x03000000	Flood to the VLAN	Same as non-MCT switch

Port loop detection

Port loop detection is used to detect L2 loops in MCT (due to misconfiguration). When using MCT, it requires the ICL ports to be strictly tagged. The port loop detection feature supports strictly tagged ports.

Loop detection for specific VLAN on a port

Strict mode loop detection can be configured on a specific VLAN for a given port. To configure loop detection on VLAN 10 for interface 1/1, enter a command such as the following.

```
NetIron(config-if-e1000-1/1)#loop-detection vlan 10
```

Syntax: [no] loop-detection [vlan <vlan_id>]

Where vlan-id enablesLoose Mode configuration for a VLAN group.

A port can be tagged or untagged member of this VLAN.

Multiple VLANs can have loop detection configured for a given port. Loop detection BPDUs will be sent out of each configured VLAN on that port.

Loop detection shutdown-disable

Use the loop-detection shutdown-disable command to disable the port shutdown feature in case of loop detection. This feature will ensure that the ICL stays up when a loop detection PDU is received on the ICL. This command will be applied to both strict mode or loose mode loop detection. To configure loop-detection shutdown-disable to shutdown port 1/1 used for the ICL link, enter a command such as the following.

```
NetIron(config-if-e1000-1/1)#loop-detection shutdown-disable
```

Syntax: loop-detection shutdown-disable

Loop-detection shutdown-sending-port

By default, the receive-port is shutdown by loop detection. The loop-detection shutdown-sending-port command will shutdown the port that sent the loop detection PDUs instead of shutting down the receiving port. This will ensure that the ICL stays up when a loop detection PDU is received on the ICL.

This feature is only applicable to strict mode loop detection.

```
NetIron(config-if-e1000-1/1)#loop-detection shutdown-sending-port
```

Syntax: [no] loop-detection shutdown-sending-port

Loop-detection-syslog-duration

If any of the ports has shutdown disabled, any loop detection will be logged into the syslog. Since the port is not shutdown, loop detect PDUs will come at a very fast rate and entries into the syslog are throttled.

By default, syslog-duration is 10 minutes. The configurable range is from 10 minutes to 1440 minutes.This is a global command and any changes will be applied to all interfaces.To configure loop-detection-syslog-duration for every 30 minutes, enter a command such as the following.

```
NetIron(config)# loop-detection-syslog-duration 30
```

Syntax: [no] loop-detection-syslog-duration <mins>

The <mins> parameter specifies the configurable range which is from 10 minutes to 1440 minutes.

MCT failover scenarios

1. ICL interface or CCP goes down (Keep alive configured)

   When the keepalive VLAN is used and finds the cluster nodes reachability when the ICL or CCP goes down. If the peer node is reachable over keepalive VLAN, the MCT nodes perform performs the Master/Slave negotiation per client. After negotiation, the Slave shuts down its client ports whereas the Master client ports continue to forward the traffic.

   The Master/Slave negotiation is done per MCT client on the basis of RBridge Id and client Local or Remote reachability. If the client is reachable from both MCT nodes, the higher RBridge Id becomes the Master. If client is reachable from one of the MCT nodes, only then the node on which it is reachable becomes the Master.

   If the peer is not reachable over the keepalive VLAN, then both cluster nodeswill keep forwarding.

   NOTE
   Dell recommends to use keepalive VLANs with the MCT configurations. This will provide a backdoor reachability ifthe ICL interface goes down.

2. ICL interface or CCP goes down (Keep alive not configured)

   When the keepalive VLAN is not configured, both cluster nodes will keep forwarding. Use the **client-isolation strict** to remove the client interface as soon as ICL goes down and isolate the client completely

3. MCT node goes down.

   When the MCT nodes goes down, the traffic will failover to the other MCT node.

4. Hitless failover performed on one of the MCT nodes

   Traffic is switched over to the other node. However, the CCP will go down and come back up again once the hitless failover is completed.

   Use the **client-interfaces shutdown** command to shutdown all the client interfaces so that the traffic failovers to the other MCT node first. Then perform the hitless failover.

5. Client interface on one of the MCT node goes down

   When hitless failover happens on a NetIron MLX node, that node flushes all the MACs and will reestablish cluster CCP session. In this case, the user may notice some traffic impact.

6. Double failures –The ICL goes down and client interface goes down on one MCT node.

   Multiple failures could drop traffic in this scenario even if there is actual physical path is available.

Show commands

Use the **show cluster** command to display the peer and client states.

```
NetIron#show cluster

Cluster CLUSTER-1 2000
==================
Rbridge Id: 35535, Session Vlan: 2001, Keep-Alive Vlan: 201
Cluster State: Deploy
Client Isolation Mode: Loose
Configured Member Vlan Range: 2 to 2000 2002 to 4090
Active Member Vlan Range: 2 to 3 21 to 148 201 404 to 445 501 to 508 2010 3511 to
3574 4021 to 4025 4051 4070 4080 4087 4090

 ICL Info:
 ---------
 Name            Port  Trunk
 ICL-1           2/1   6

 Peer Info:
 ----------
 Peer IP: 1.1.1.1, Peer Rbridge Id: 1, ICL: ICL-1
 KeepAlive Interval: 50 , Hold Time: 300, Fast Failover
 Active Vlan Range:  2 to 3 21 to 148 201 404 to 445 501 to 508 2010 3511 to 3574
4021 to 4025 4051 4070 4080 4087 4090
 Peer State: CCP Up (Up Time:   0 days:19 hr:24 min: 8 sec)

 Client Info:
 ------------
 Name            Rbridge-id Config     Port  Trunk FSM-State
 Client1         2222       Deployed   1/2     3   Up
 Client2         222        Deployed   1/40    -   Up
```

Syntax: show cluster

Use the **show cluster client** command to display additional State Machine information including the reason for Local CCEP down.

```
NetIron#show cluster mct client c2
…
State: Remote Up
Reason for Local CCEP down: "client-interfaces shutdown" command
Number of times Local CCEP down: 2
Number of times Remote CCEP down: 1
Number of times Remote Client undeployed: 1
Total CCRR packets sent: 12
Total CCRR packets received: 13
```

Syntax: show cluster client

Following reasons are displayed for Local CCEP down.

TABLE 104 Reason for Local CCEP down

Reason for Local CCEP down	means....
client-interfaces shutdown	command is configured
client-isolation strict	command is configured

TABLE 104 Reason for Local CCEP down

Reason for Local CCEP down	means....
Deploy mismatch	Client is not deployed remotely
Slave state	Client is in Slave State when CCP is down
cluster and client undeployed	Neither the Cluster or Client is deployed.
cluster undeployed	Cluster is not deployed
client undeployed	Client is not deployed

Syslogs and debugging

Syslogs are displayed when remote CCEP is state is changed or remote client is deployed or undeployed.

```
SYSLOG: Jun 1 15:43:36:<14>Jun 1 15:43:36 CES, CLUSTER FSM: Cluster mct (Id: 1),
client c2 (RBridge Id: 4) -
    Remote client deployed
SYSLOG: Jun 1 16:04:24:<14>Jun 1 16:04:24 CES, CLUSTER FSM: Cluster mct (Id: 1),
client c2 (RBridge Id: 4) -
    Remote client CCEP up
```

Sample configuration

The output below is a sample configuration using port loop detection.

```
NetIron#show run
lag "icl1" dynamic id 1
ports ethernet 3/20 ethernet 4/9
primary-port 3/20
deploy
!
vlan 1 name DEFAULT-VLAN
no untagged ethe 3/20 ethe 4/9
!
vlan 10
tagged ethe 3/20 ethe 4/9
router-interface ve 10
!
vlan 11
untagged ethe 4/17
tagged ethe 3/11 ethe 3/20 ethe 4/9
loop-detection
!
vlan 15
tagged ethe 3/20 ethe 4/9
!
vlan 20
tagged ethe 3/11 ethe 3/20 ethe 4/9 ethe 4/17
!
no route-only
logging console
telnet server
loop-detection-interval 1
loop-detection-disable-duration 1
```

```
loop-detection-syslog-duration 11

!
interface ethernet 3/20
loop-detection shutdown-disable
loop-detection vlan 20
!
interface ethernet 4/17
enable
!
loop-detection shutdown-sending-port
loop-detection vlan 20
loop-detection vlan 11
!
interface ve 10
ip address 10.10.10.1/24
!
!
!
cluster abc 1
rbridge-id 100
session-vlan 10
keep-alive-vlan 15
member-vlan 11 to 20
member-vlan 40 to 50
icl icl1 ethernet 3/20
peer 10.10.10.2 rbridge-id 200 icl icl1
client c1
  rbridge-id 300
  client-interface ethernet 3/11
!
```

MAC Database Update (MDUP)

The MACs that are learned locally are given the highest priority or the cost of 0 so they are always selected as best MAC.

Each MAC is advertised with a cost. Low cost MACs are given preference over high cost MACs.

If a MAC moves from a CCEP port to a CEP port, a MAC move message is sent to the peer and the peer moves the MAC from its CCEP ports to the ICL links.

If the cost of a MAC is the same, then the MAC learned from the Lower RBridge ID wins and is installed in the FDB.

Cluster Mac types

Cluster Local MAC (CL): MACs that are learned on VLANs that belongs to cluster VLAN range and on CEP ports locally.

MACs are synchronized to the cluster peer and are subject to aging.

```
NetIron#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/1 0 20 CL Default_ESI
```

Cluster Remote MAC (CR): MACs that are learned via MDUP message from the peer (CL on the peer) The MACs are always programmed on the ICL port and do not age. They are deleted only when it is deleted from the peer. A MDB entry is created for these MACs with a cost of 1, and associated with the peer rbridge id.

```
NetIron#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CR Default_ESI
```

Cluster Client Local MAC (CCL): MACs that are learned on VLANs that belongs to cluster VLAN range and on CCEP ports.

The MACs are synchronized to the cluster peer and are subject to aging. A MDB entry is created for these MACs with a cost of 0 and are associated with the client and cluster rbridge ids.

```
NetIron#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCL Default_ESI
```

Cluster Client Remote MAC (CCR): MACs that are learned via MDUP message from the peer (CCL on the peer) The MACs are always programmed on the corresponding CCEP port and do not age. They are deleted only when it is deleted from the peer. A MDB entry is created for the MACs with the cost of 1, and are associated with the client and peer rbridge ids.

```
NetIron#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCR Default_ESI
```

MAC aging

Only the local MAC entries are aged on a node. The remote MAC entries will be aged based on explict MDUP messages only.

The remote MACs learned through MDUP messages are dynamic MACs with the exception that they never age from FDB.

MAC flush

If the CEP port is down, the MACs are flushed and individual MAC deletion messages are sent to the Peer.

If the CCEP local port is down, the MACs are flushed locally and individual MAC deletion messages are sent to peer.

If the **clear mac** command is given, all the MDB and FDB are rebuilt.

If the **clear mac vlan** command is given, all the local MDB and FDB are rebuilt for that VLAN.

MAC movement happens normally on the local node.

CEP to CCEP MAC movement – MAC movement normally happens on the local node, and deletes all the other MDBs from the peer to create a new local MDB.

CCEP to CEP MAC movement - MAC movement happens normally on the local node and delete all the other MDBs from the peer to create a new local MDB.

Flooding support on VLANs

Dell support the existing VLAN hardware flooding features such as unknown-unicast-flooding and vlan-cpu-protection on cluster VLANs. However, some changes were made to the way CAM entries are programmed. To support MCT cluster VLANs, the following changes to how the CAM is programmed:

- If the ICL port is part of a PPCR, then the device will program the specific (port or VLAN) based hardware flooding CAM entries on that PPCR. This is to avoid duplicate hardware flooding packets to be sent to CCEP ports.
- On an ICL port, the FID in pram will point to MCT_VLAN_CCEP_CONTROL_FID
- On non-ICL port, the FID in pram will point to VLAN_FID

Show Commands

To display all MAC entries, use the **show mac** command as shown below:

```
NetIron# show mac
Total active entries from all ports = 120000
Type Code - ST:Static SEC:Secure 1x:Dot1x NA: NotAvail A:Allow D:Deny
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
Port Type - CEP:Customer Edge PNP:Provider Network BEP:Backbone Edge
BNP:Backbone Network
Vlan Type - C:Customer S:Service B:Backbone I:ISID
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0101.84ff 1/13 0 20 CCL Default_ESI
0000.0100.4780 1/13 0 20 CCR Default_ESI
0000.0800.0663 1/14 0 20 CL Default_ESI
0000.0800.0870 1/1 0 20 CR Default_ESI
```

Syntax: show mac

To display all the Cluster Local MAC entries for a cluster, use the **show mac cluster** command as shown below:

```
NetIron#show mac cluster abc
Total Cluster Enabled(CL+CR+CCL+CCR) MACs: 451
Total Cluster Local(CL) MACs: 100
Total Cluster Remote(CR) MACs: 151
Total Cluster Client Macs(CCL+CCR) for all clients: 200
Total Cluster Client Local(CCL) MACs for all clients: 200
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCL Default_ESI
0000.0800.3500 1/13 0 20 CCR Default_ESI
```

Syntax: show mac [cluster <id>|<name> <local>| <remote>]

Clear MAC commands

To clear all MACs in the system, enter a command such as the following.

```
NetIron#clear mac
```

Syntax: clear mac

Clear cluster specific MACs

To clear cluster specific MACs in the system, enter a command such as the following.

```
NetIron#clear mac cluster TOR 1 local
```

Syntax: clear mac cluster <cluster-id> | <cluster-name> { local | remote }

Clear client specific MACs

To clear client specific MACs in the system, enter a command such as the following.

```
NetIron#clear mac cluster TOR 1 client 1 local
```

Syntax: clear mac cluster <cluster-id> | <cluster-name> client <client-name> { local | remote }

Clear VLAN specific MACs

To clear VLAN specific MACs in the system, enter a command such as the following.

```
NetIron#clear mac vlan 2
```

Syntax: clear mac vlan <vlan_id>

Clear cluster VLAN specific MACs

To clear cluster VLAN specific MACs in the system, enter a command such as the following.

```
NetIron#clear mac cluster cluster TOR 1 vlan 1 local
```

Syntax: clear mac cluster <cluster_id> | <cluster-name> vlan <vlan_id> {Local | Remote}

Clear cluster client vlan specific MACs

To clear cluster client specific MACs in the system, enter a command such as the following.

```
NetIron#clear mac cluster TOR 1 vlan 2 client client 1 local
```

Syntax: clear mac cluster <cluster_id> |< cluster-name> vlan <vlan_id> client <client_name> {Local | Remote}

Displaying MDUP packet statistics

To display the statistics of MDUP packets, enter a command such as the following.

```
NetIron#show mac mdup-stats
MDUP Information
================
MDUP Data buffers in queue : 0
MDUP Statistics
===============
MDUP Update Messages sent: 7
Add Mac sent: 20
Del Mac sent: 0
Move Mac sent: 0
MDUP Mac Info Messages sent: 1
MDUP Flush Messages sent: 1
MDUP Synch Messages sent: 0
MDUP Update Messages received: 3
Add Mac received: 40
Del Mac received: 0
```

```
Move Mac received: 0
MDUP Mac Info Messages received: 0
MDUP Flush Messages received: 0
MDUP Synch Messages received: 0
```

Syntax: show mac mdup-stats

Clearing the statistics of MDUP packets

To clear the statistics of MDUP packets, enter a command such as the following.

```
NetIron# clear mac mdup-stats
```

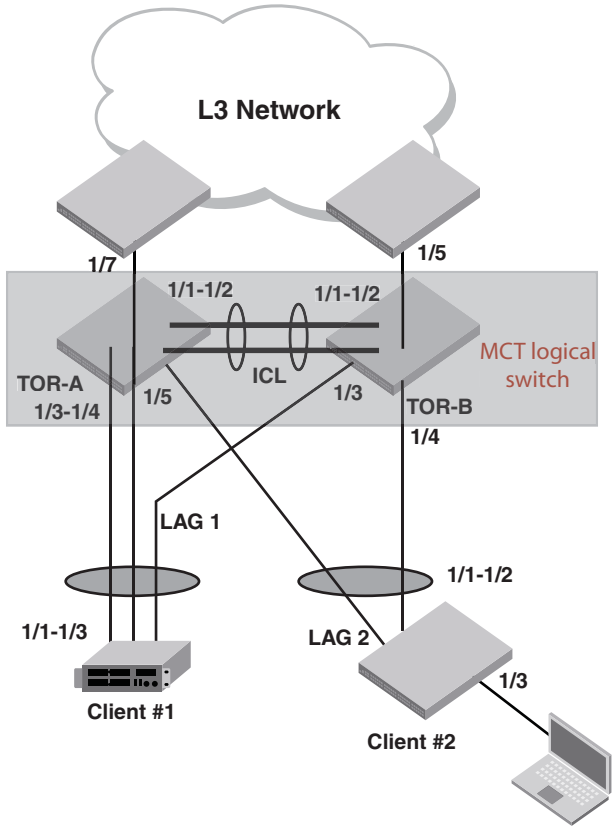
Syntax: clear mac mdup-stats

MCT configuration examples

The following examples displays the module provisioning information from a configuration file:

Single level MCT example

FIGURE 113 Single level MCT



TOR-A:

```
lag "1" dynamic id 1
```

```
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-B:1/1" ethernet 1/1
 port-name "ICL-to-TOR-B:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/1" ethernet 1/3
 port-name "lag-client-1:1/2" ethernet 1/4
!
lag "3" dynamic id 3
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-2:1/1" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/7
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 port-name CEP-PC
 enable
!
interface ethernet 1/7
 port-name to-L3-ECMP
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
```

```
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/5
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-A:1/1" ethernet 1/1
 port-name "ICL-to-TOR-A:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/3" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4
 primary-port 1/4
 deploy
 port-name "lag-client-2:1/2" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
```

```
interface ethernet 1/5
 port-name to-L3-ECMP
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

Client-1:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/3
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-A" ethernet 1/2
 port-name "lag-to TOR-B" ethernet 1/3
!
interface ethernet 1/1
 enable
!
end

----------------------------------------
```

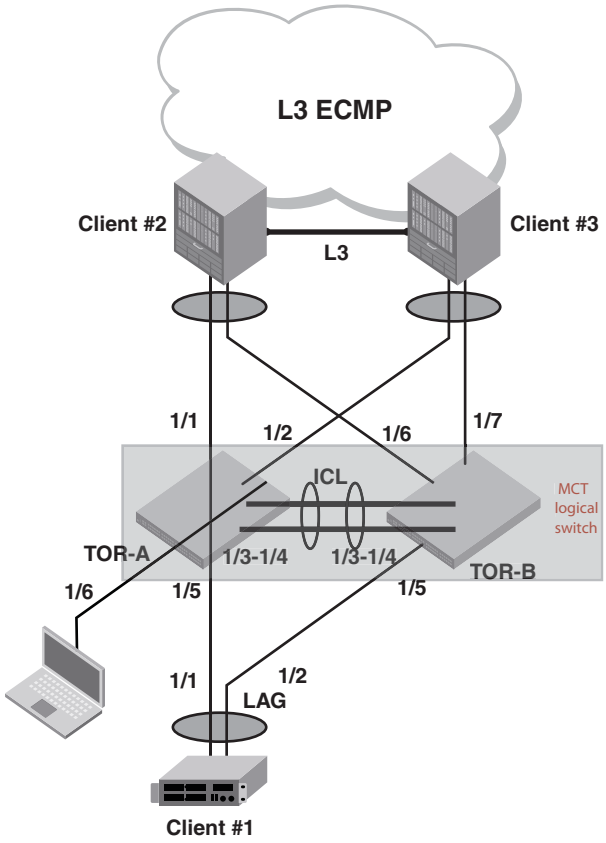
Client-2:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name to-Host-PC
 enable
```

```
!
end
```

Single level MCT- extension example

FIGURE 114 Single level MCT- extension



TOR-A:

```
lag "1" dynamic id 1
 ports ethernet 1/1
 primary-port 1/1
 deploy
 port-name "lag-client-2:1/1" ethernet 1/1
!
lag "2" dynamic id 2
 ports ethernet 1/2
 primary-port 1/2
 deploy
 port-name "lag-client-3:1/1" ethernet 1/2
!
lag "3" dynamic id 3
 ports ethernet 1/3 to 1/4
 primary-port 1/3
```

```
 deploy
 port-name "ICL-to-TOR-B:1/3" ethernet 1/3
 port-name "ICL-to-TOR-B:1/4" ethernet 1/4
!
lag "4" dynamic id 4
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-1:1/1" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/3 to 1/4
!
vlan 2 name client-VLAN
 untagged ethe 1/1 to 1/2 ethe 1/5 to 1/6
 tagged ethe 1/3 to 1/4
!
vlan 4090 name Session-VLAN
 tagged ethe 1/3 to 1/4
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/2
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 port-name CEP-PC
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/3
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/5
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/1
  deploy
```

```
 client Client-3
  rbridge-id 300
  client-interface ethernet 1/2
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/6
 primary-port 1/6
 deploy
 port-name "lag-client-2:1/2" ethernet 1/6
!
lag "2" dynamic id 2
 ports ethernet 1/7
 primary-port 1/7
 deploy
 port-name "lag-client-3:1/2" ethernet 1/7
!
lag "3" dynamic id 3
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "ICL-to-TOR-A:1/3" ethernet 1/3
 port-name "ICL-to-TOR-A:1/4" ethernet 1/4
!
lag "4" dynamic id 4
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-1:1/2" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/3 to 1/4
!
vlan 2 name client-VLAN
 untagged ethe 1/5 to 1/7
 tagged ethe 1/3 to 1/4
!
vlan 4090 name Session-VLAN
 tagged ethe 1/3 to 1/4
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 enable
!
interface ethernet 1/7
```

```
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/3
 peer 1.1.1.1 rbridge-id 1 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/5
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/6
  deploy
 client Client-3
  rbridge-id 300
  client-interface ethernet 1/7
  deploy
!
end
----------------------------------------
```

Client-1:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
interface ethernet 1/1
 enable
!
end
----------------------------------------
```

Client-2:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
vlan 2
 untagged ethe 1/1 to 1/3
 router-interface ve 2
!
router ospf
 area 0
```

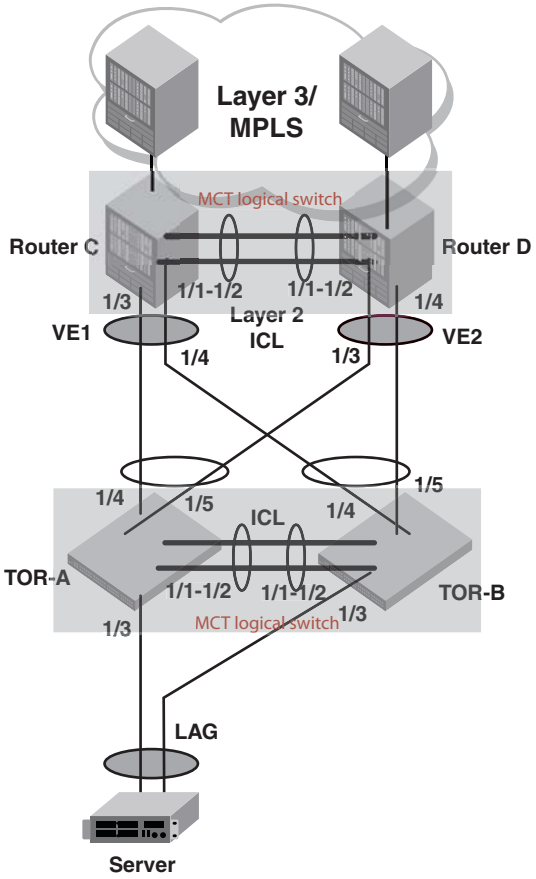
```
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name L3-ECMP-Cloud
!
interface ve 2
 ip address 10.10.10.1/24
 ip ospf area 0
!
end
----------------------------------------
```

Client-3:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
vlan 2
 untagged ethe 1/1 to 1/3
 router-interface ve 2
!
router ospf
 area 0
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name L3-ECMP-Cloud
!
interface ve 2
 ip address 10.10.10.2/24
 ip ospf area 0
!
end
```

Two level MCT example

TABLE 105 Two level MCT



TOR-A:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-B:1/1" ethernet 1/1
 port-name "ICL-to-TOR-B:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
 deploy
 port-name "lag-client-Server:1" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4 to 1/5
 primary-port 1/4
 deploy
 port-name "lag-Router-C:1/3" ethernet 1/4
 port-name "lag-Router-D:1/3" ethernet 1/5
```

```
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Server-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Routers
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-A:1/1" ethernet 1/1
 port-name "ICL-to-TOR-A:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
```

```
 deploy
 port-name "lag-client-Server:2" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4 to 1/5
 primary-port 1/4
 deploy
 port-name "lag-Router-C:1/4" ethernet 1/4
 port-name "lag-Router-D:1/4" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl TOR
 deploy
 client Server-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Routers
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

Router C:

```
lag "1" dynamic id 1
```

```
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Router-D:1/1" ethernet 1/1
 port-name "ICL-to-Router-D:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-TOR-A:1/4" ethernet 1/3
 port-name "lag-TOR-B:1/4" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MPLS-Cloud
 enable
!
interface ve 100
 ip address 1.1.1.3/24
!
!
cluster Router 2
 rbridge-id 3
 session-vlan 4090
 member-vlan 2
 icl Router ethernet 1/1
 peer 1.1.1.4 rbridge-id 4 icl Router
 deploy
 client TOR
  rbridge-id 1
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

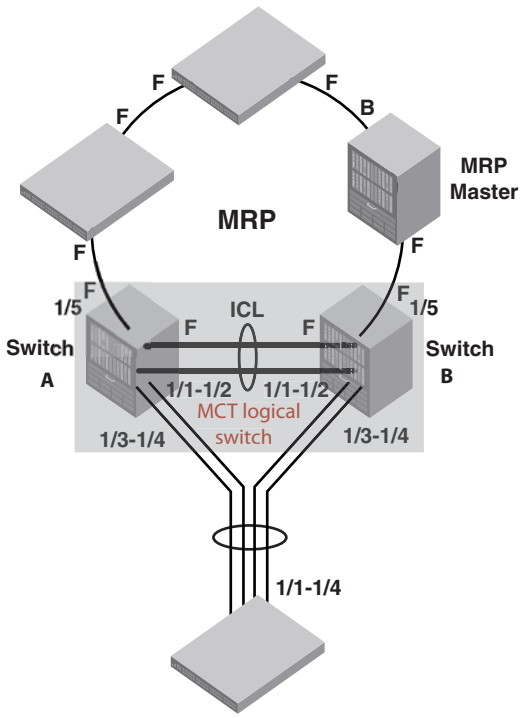
Router D:

```
lag "1" dynamic id 1
```

```
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Router-C:1/1" ethernet 1/1
 port-name "ICL-to-Router-C:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-TOR-A:1/5" ethernet 1/3
 port-name "lag-TOR-B:1/5" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MPLS-cloud
 enable
!
interface ve 100
 ip address 1.1.1.4/24
!
!
cluster Router 2
 rbridge-id 4
 session-vlan 4090
 member-vlan 2
 icl Router ethernet 1/1
 peer 1.1.1.3 rbridge-id 3 icl Router
 deploy
 client TOR
  rbridge-id 1
  client-interface ethernet 1/3
  deploy
!
end
```

MRP integration with MCT example

FIGURE 115 MRP integration with MCT



MCT-capable-switch-A

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-2:1/1" ethernet 1/1
 port-name "ICL-to-Switch-2:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/1" ethernet 1/3
 port-name "lag-client-1:1/2" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
 metro-ring-1
  ring-interfaces ethe 1/1 ethe 1/5
  enable
```

```
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname Switch-1
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MRP-from-Master
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster MRPRing 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl MRPRing ethernet 1/1
 peer 1.1.1.2 rbridge-id 2 icl MRPRing
 deploy
 client client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

MCT-capable-switch-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-1:1/1" ethernet 1/1
 port-name "ICL-to-Switch-1:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/3" ethernet 1/3
 port-name "lag-client-1:1/4" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
```

```
 metro-ring-1
  ring-interfaces ethe 1/1 ethe 1/5
  enable
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname Switch-2
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MRP-to-Master
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster MRPRing 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl MRPRing ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl MRPRing
 deploy
 client client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

client-Switch:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/4
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-1:1/3" ethernet 1/1
 port-name "ICL-to-Switch-1:1/4" ethernet 1/2
 port-name "ICL-to-Switch-2:1/3" ethernet 1/1
 port-name "ICL-to-Switch-2:1/4" ethernet 1/2
!
interface ethernet 1/1
 enable
!
end
```

Multi-Chassis Trunk (MCT) for VRRP or VRRP-E

One MCT switch is the VRRP or VRRP-E master router and the other MCT switch is VRRP or VRRP-E backup router

The MCT switch that acts as backup router needs to ensure that packets sent to a VRRP-E virtual IP address can be L2 switched to the VRRP-E master router for forwarding. The MCT switch that acts as master router will sync the VRRP-E MAC to the other MCT switch that acts as backup router. Both data traffic and VRRP-E control traffic travel through the ICL unless the short-path forwarding feature is enabled.

L3 traffic forwarding from CEP ports to CCEP ports

Traffic destined to the CCEP ports from the client or CEP ports follow the normal IP routing on both master and backup routers. By default, the best route should not involve the ICL link. Only when the direct link from CEP ports to CCEP ports are down will the traffic be re-routed to pass through ICL link.

ARP broadcast resolution

Assuming that switch A is VRRP-E master router and switch B is the backup router. ARP request (a broadcast packet) from S1 that is sent through direct link to switch B will be sent to switch A for processing through ICL link. Since MAC learning is disabled on ICL link, the ARP will not be learned automatically through the ICL link. When the ARP request is received by switch A, the reply will be sent through direct link from switch A to S1. If by the time the ARP reply was received the MAC address for the MCT on S1 is not learned yet, the reply packet may be flooded to both the CCEP ports and ICL ports.

Both MCT switches are VRRP or VRRP-E backup routers

In Figure 116, both MCT switches C and D need to ensure packets sent to VRRP-E virtual IP address can be L2 switched to the VRRP-E master router for forwarding. The MCT switch that has direct connection to the master router (who actually learned the VRRP-E MAC from the master) will sync the VRRP-E MAC to the other MCT switch that does not have direct connection to the master. Both data traffic and VRRP-E control traffic travel through ICL unless the short-path forwarding feature is enabled.

FIGURE 116 Example of MCTS that are Layer 2 switched

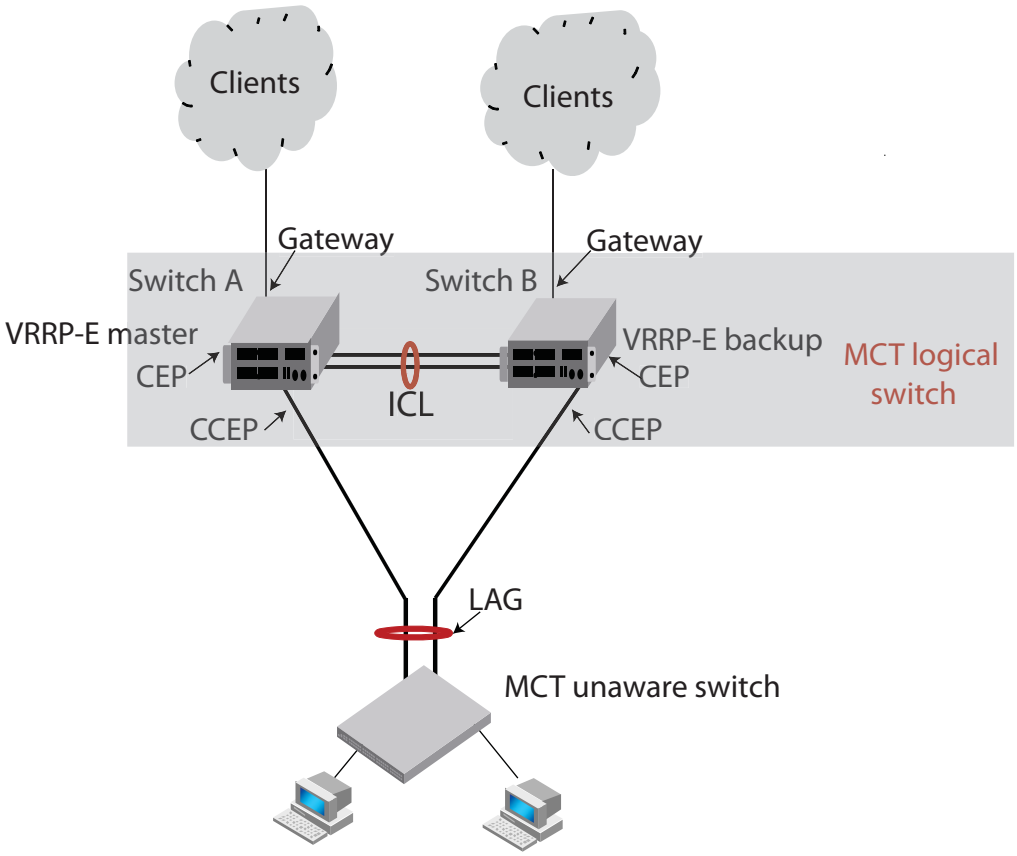
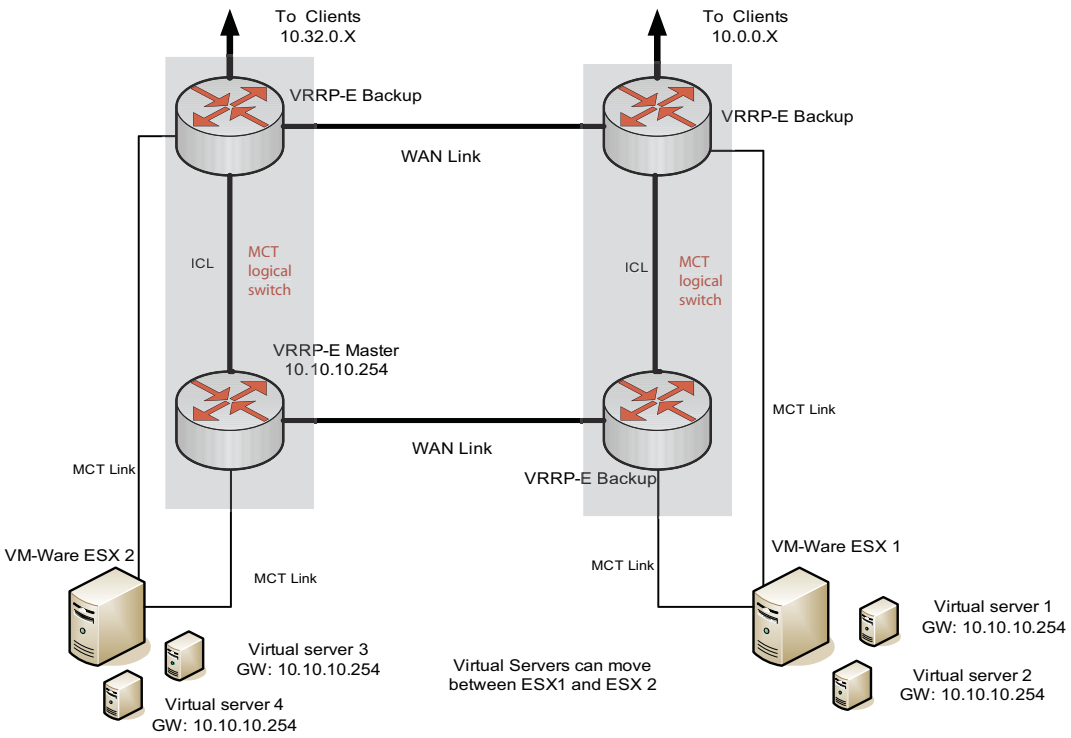


FIGURE 117 Example of MCTs that are deployed on two sites that are connected through two WAN links.



In Figure 117, MCTs are deployed on two sites that are connected through two WAN links.

- Two WAN links are completely independent. Switch A and B form MCT 1 and switch C and D form MCT 2. There are L2 protocols running on the VRRP-E routers. L2 protocols will block one of the WAN links to ensure loop-free topology.

Configuration considerations

- VRRE-E virtual MAC will be synced and learned on ICL ports on backup routers through the ICL.
- VRRP or VRRP-E master router will be broadcast hello packets to all VLAN member ports including ICL ports. Normal VLAN FID will be used for broadcasting.
- VRRP or VRRP-E backup routesr will not be flood back hello packets received from ICL ports to ICL ports, but will be flooded to other non- ICL ports.
- In the current release, MCT switches must have complete routing information using static routes for L3 forwarding.
- For MCT switches configured with VRRP or VRRP-E, track-port features can be enabled to track the link status to the core switches so the VRRP or VRRP-E failover can be triggered.

NOTE
Dell recommends disabling ICMP redirect globally to avoid unintended CPU forwarding of traffic when VRRP or VRRP-E is configured.

L3 traffic forwarding behaviors

When one MCT switch act as VRRP or VRRP-E master router and the other MCT switch is VRRP or VRRP-E backup, the following behavior will be seen:

- Packets sent to VRRP-E virtual IP address will be L2 switched to the VRRP-E master router for forwarding.
- The VRRP-E MAC will be learned by the other MCT switch that acts as backup router.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

When both MCT devices act as the VRRP or VRRP-E backup routers, the following behavior will be seen:

- Packets sent to VRRP-E virtual IP address will be L2 switched to the VRRP-E master router for forwarding.
- VRRP-E MAC will be learned by both MCT switches acting as backup routers.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

VRRP-E short-path forwarding and revertible option

The **track-port** command will monitor the status of the outgoing port on the backup. It will revert back to standard behavior (no short-path forwarding) temporarily even if short-path forwarding is configured.

Under the VRRP-E VRID configuration level, use the **short-path-forwarding** command. If the revertible option is not enabled, the default behavior will remain the same. Use the following command to enable short path forwarding.

```
NetIron(config-if-e1000-vrid-2)#short-path-forwarding revert-priority 60
```

Syntax: [no] short-path-forwarding [revert-priority <value>**]**

Use the supplied priority value as a threshold to determine if the **short-path-forwarding** behavior should be effective or not. If one or more ports tracked by the **track-port** command go down, the current priority of VRRP-E will be lowered by a specific amount configured in the **track-port** command for each port that goes down.

Once the current-priority is lower than the threshold, the **short-path-forwarding** will be temporally suspended and revert back to the regular VRRP-E forwarding behavior without **short-path-forwarding** enabled.

The reverting behavior is only temporary. If one or more of the already down ports tracked by the **track-port** command come back, it is possible that the current priority of VRRP-E will be higher than the threshold again and the **short-path-forwarding** behavior will be resumed.

Configuring IP 19

Overview

The following IP features are supported by the NetIron MLX Series devices.

- GRE IP Tunnel
- Multicast over GRE Tunnel
- ICMP Error Message Rate Increase
- Restart Global Timers
- Restart helper-mode
- ARP Inspection
- DHCP Snooping
- DHCP Relay Enhancement
- DNS queries of IPv4 DNS Servers
- DNS queries of IPv6 DNS Servers
- IRDP
- UDP Broadcast and IP Helper
- Configuring BootP/DHCP Forwarding Parameters
- IPv6 Over IPv4 Tunnels in Hardware
- Disabling Gratuitous ARP Requests for Local Proxy ARP
- Dynamic ARP Inspection (DAI)
- DHCP Option 82 insertion
- IP Source Guard
- Displaying the IP Route Table
- Jumbo Frames
- IPv4 VRF Support
- Configuring a Static IP Route between VRFs
- Naming a Static IP Route
- New minimum GRE keepalive
- Dropping Traffic Sent to the Null0 Interface in Hardware
- CAM Default Route Aggregation
- Static route to an LSP Tunnel Interface
- Statistics for GRE and Manual IPv6 Tunnels
- Multi-port static ARP

Internet Protocol (IP) is enabled by default. This chapter describes how to configure IP parameters on the PowerConnect.

Basic configuration consists of adding IP addresses and enabling a route exchange protocol. Refer to "Configuring IP addresses" on page 674.

To change some of the IP parameters from their default values or to view configuration information or statistics, refer to the following sections:

- "The IP packet flow" on page 650
- "Basic IP parameters and defaults" on page 654
- "Configuring IP parameters" on page 674

The IP packet flow

Figure 118 shows how an IP packet moves through a PowerConnect.

FIGURE 118 IP Packet flow through a PowerConnect

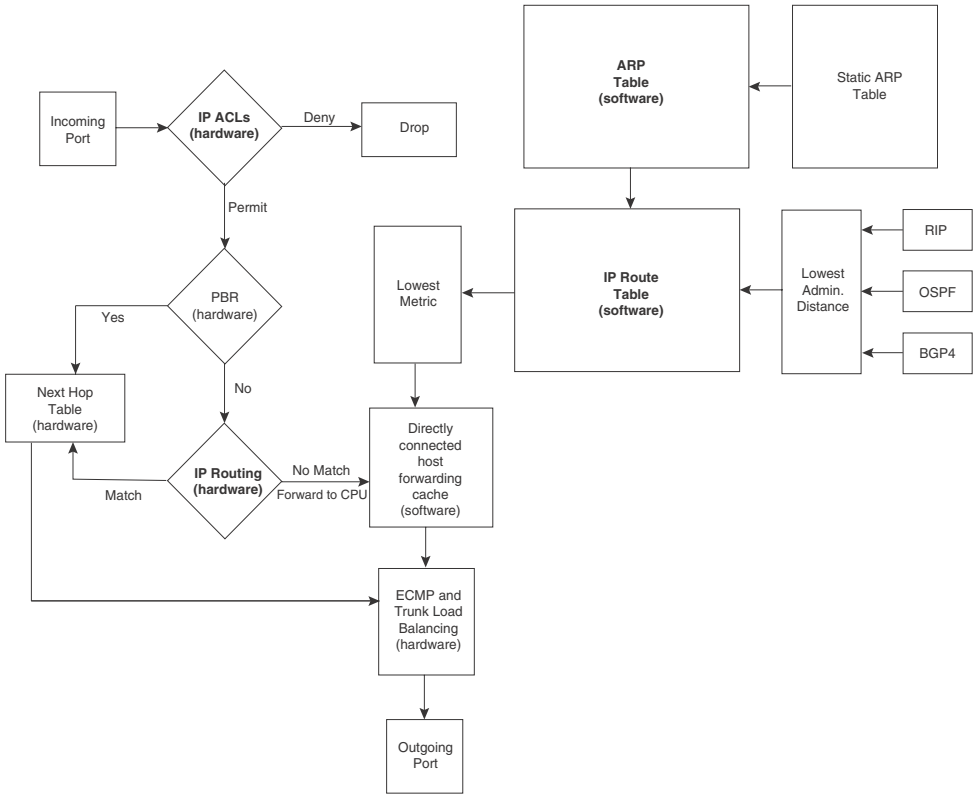


Figure 118 shows the following packet flow.

1. When the PowerConnect receives an IP packet, the PowerConnect checks for IP ACL filters on the receiving interface. If a deny filter on the interface denies the packet, the PowerConnect discards the packet and performs no further processing. If logging is enabled for the filter, then the PowerConnect generates a Syslog entry and SNMP trap message.
2. If the packet is not denied, the PowerConnect checks for Policy Based Routing (PBR). If the packet matches a PBR policy applied on the incoming port, the PBR processing is performed and either drops the packet or forwards it to a port, based on the route map rules.
3. If the incoming packet does not match PBR rules, the PowerConnect looks in the hardware IP routing table to perform IP routing. The hardware routing table is pre-loaded with the complete routing table, except for the directly connected host entries. Default and statically defined routes are also pre-loaded in the hardware routing table. If the incoming packet matches a route entry, the packet is routed according to the information provided in the route entry. The ECMP and LAG load balancing is done by the hardware, if needed, to select the outgoing port.
4. If there is no match in the IP routing table and a default route is not configured, the packet is dropped. For an IP packet whose destination IP address is to a directly connected host, the first packet is forwarded to the CPU. If the ARP is resolved and the host is reachable, the CPU creates a route entry in the hardware to route subsequent packets in hardware.

The software enables you to display the ARP cache and static ARP table, the IP route table, the IP forwarding cache.

You also can change the capacity of the following tables by changing the memory allocation for the table:

- "ARP cache table" on page 651
- "Static ARP table" on page 652
- "IP route table" on page 652
- "IP forwarding cache" on page 653

ARP cache table

The Address Resolution Protocol (ARP) is supported on the PowerConnect. Refer to "Configuring ARP parameters" on page 686.

The ARP cache contains entries that map IP addresses to MAC addresses. Generally, the entries are for devices that are directly attached to the PowerConnect.

An exception is an ARP entry for an interface-based static IP route that goes to a destination that is one or more router hops away. For this type of entry, the MAC address is either the destination device's MAC address or the MAC address of the router interface that answered an ARP request on behalf of the device, using proxy ARP.

The ARP cache can contain dynamic (learned) entries and static (user-configured) entries. The software places a dynamic entry in the ARP cache when the PowerConnect learns a device's MAC address from an ARP request or ARP reply from the device.

The software can learn an entry when the PowerConnect receives an ARP request from another IP forwarding device or an ARP reply.

Example : Dynamic entry

```
      IP Address          MAC Address         Type        Age     Port
1     207.95.6.102        0800.5afc.ea21      Dynamic     0         6
```

Each entry contains the destination device's IP address and MAC address.

Static ARP table

In addition to the ARP cache, the PowerConnect has a static ARP table.

Entries in the static ARP table are user-configured. You can add entries to the static ARP table regardless of whether the device the entry is for is connected to the PowerConnect.

The software places an entry from the static ARP table into the ARP cache when the entry's interface comes up.

Example : Static ARP entry

```
  Index   IP Address          MAC Address         Port
  1       207.95.6.111        0800.093b.d210      1/1
```

Each entry lists the information you specified when you created the entry.

To display ARP entries, refer to the following:

- "Displaying the ARP cache" on page 759
- "Displaying the static ARP table" on page 760

To configure other ARP parameters, refer to "Configuring ARP parameters" on page 686.

To increase the size of the ARP cache and static ARP table, refer to the following:

- For dynamic entries, refer to the "Displaying and modifying default settings for system parameters" on page 100. The ip-arp parameter controls the ARP cache size.

IP route table

The IP route table contains paths to IP destinations.

The IP route table can receive the paths from the following sources:

- A directly-connected destination, which means there are no router hops to the destination
- A static IP route, which is a user-configured route
- A route learned through RIP
- A route learned through OSPF
- A route learned through ISIS
- A route learned through BGP4

The IP route table contains the best path to a destination:

- When the software receives paths from more than one of the sources listed above, the software compares the administrative distance of each path and selects the path with the lowest administrative distance. The administrative distance is a protocol-independent value from 1 – 255.
- When the software receives two or more best paths from the same source and the paths have the same metric (cost), the software can load share traffic among the paths based on Layer 2, Layer 3 and TCP/UDP information.

Example : IP route table

```
Destination       NetMask           Gateway           Port   Cost   Type
1.1.0.0           255.255.0.0       99.1.1.2          1/1    2      R
```

Each IP route table entry contains the destination's IP address and subnet mask and the IP address of the next-hop router interface to the destination. Each entry also indicates the port attached to the destination or the next-hop to the destination, the route's IP metric (cost), and the type. The type indicates how the IP route table received the route.

To display the IP route table, refer to "Displaying the IP route table" on page 763.

To configure a static IP route, refer to "Configuring static routes" on page 714.

To clear a route from the IP route table, refer to "Clearing IP routes" on page 767.

To increase the size of the IP route table for learned and static routes, refer to "Displaying and modifying default settings for system parameters" on page 100.

Consider the following:

- For learned routes, modify the ip-route parameter.
- For static routes, modify the ip-static-route parameter.

IP forwarding cache

The PowerConnect maintains a software cache table for fast processing of IP packets that are forwarded or generated by the CPU. The cache also contains forwarding information that is normally contained in the IP routing table. For example, the cache contains information on the physical outgoing port, priority, VLAN, and the type of cache entry. Also, cache entries have hardware information, which is useful for debugging and aging.

There are two types of IP cache entries.

1. Directly connected host entries – These entries are created when the CPU receives the first packet destined to a directly connected host. Host entries are set to age out after a certain period if no traffic is seen for that entry.
2. Network entries – These entries are created when a route table entry is created in software. These entries are not subjected to aging. A route table entry is created when routes are learned by routing protocols such as OSPF or when routes are statically configured.

Example : IP forwarding cache

```
      IP Address       Next Hop         MAC              Type  Port  Vlan  Pri
1     192.168.1.11     DIRECT           0000.0000.0000   PU    n/a         0
```

Each IP forwarding cache entry contains the IP address of the destination, and the IP address and MAC address of the next-hop router interface to the destination. If the destination is actually an interface configured on the PowerConnect itself, as shown here, then next-hop information indicates this. The port through which the destination is reached is also listed, as well as the VLAN and Layer 4 QoS priority associated with the destination if applicable.

To display the IP forwarding cache, refer to "Displaying the forwarding cache" on page 761.

IP packet queuing

When the user wants to send a packet to a local host, the software looks up the IP in the ARP cache. If the address is found, it gets the MAC address, constructs an Ethernet header with the correct source or destination MAC addresses, and sends it.

If the address is not found in the table, ARP broadcasts a packet to every host on the Ethernet, except the one from which it received the packet. The packet contains the IP address for which an Ethernet address is sought. If a receiving host identifies the IP address as its own, it will send its Ethernet address back to the requesting host.

For management of IP packet queuing when a packet is received for a directly connected host when there is no MAC address available, the **ip drop-arp-pending-packets** command has been added to allow the packets in the CPU to be dropped.

To set all packets in the LP buffer to be dropped when ARP resolution is going on, enter a command such as the folowing:

```
NetIron(config)#ip drop-arp-pending-packets
```

Syntax: [no] ip drop-arp-pending-packets

Use the **no ip drop-arp-pending-packets command** to return to the default behavior of continue with pending IP packets while ARP resolution.

Basic IP parameters and defaults

IP is enabled by default. The following protocols are disabled by default:

- Route exchange protocols (RIP, OSPF, ISIS, BGP4)
- Multicast protocols (IGMP, PIM-DM, PIM-SM, DVMRP)
- Router redundancy protocols (VRRPE, VRRP, FSRP)

When parameter changes take effect

Most IP parameters described in this chapter are dynamic. They take effect immediately, as soon as you enter the CLI command. You can verify that a dynamic change has taken effect by displaying the running configuration. To display the running configuration, enter the **show running-config** or **write terminal** command at any CLI prompt.

To save a configuration change permanently so that the change remains in effect following a system reset or software reload, save the change to the startup configuration file. Enter the **write memory** command from the Privileged EXEC level of any configuration level of the CLI.

Changes to memory allocation require you to reload the software after you save the changes to the startup configuration file. When reloading the software is required to complete a configuration change, the procedure that describes the configuration change includes a step for reloading the software.

IP global parameters

Table 106 lists the IP global parameters for the PowerConnect, their default values, and where to find configuration information.

TABLE 106 IP global parameters

Parameter	Description	Default	See page...
IP state	The Internet Protocol, version 4	Enabled **NOTE:** You cannot disable IP.	n/a
IP address and mask notation	Format for displaying an IP address and its network mask information. You can enable one of the following: • Class-based format; example: 192.168.1.1 255.255.255.0 • Classless Interdomain Routing (CIDR) format; example: 192.168.1.1/24	Class-based **NOTE:** Changing this parameter affects the display of IP addresses, but you can enter addresses in either format regardless of the display setting.	page 674
Router ID	The value that routers use to identify themselves to other routers when exchanging route information. OSPF and BGP4 use router IDs to identify routers. RIP does not use the router ID.	The IP address configured on the lowest-numbered loopback interface. If no loopback interface is configured, then the lowest-numbered IP address configured on the device.	page 684
IP Maximum Transmission Unit (MTU)	The maximum length an Ethernet packet can be without being fragmented.	1500 bytes for Ethernet II encapsulation 1492 bytes for SNAP encapsulation	page 681
Address Resolution Protocol (ARP)	A standard IP mechanism that routers use to learn the Media Access Control (MAC) address of a device on the network. The router sends the IP address of a device in the ARP request and receives the device's MAC address in an ARP reply.	Enabled	page 686
ARP rate limiting	Lets you specify a maximum number of ARP packets the device will accept each second. If the device receives more ARP packets than you specify, the device drops additional ARP packets for the remainder of the one-second interval.	Disabled	page 687
ARP age	The amount of time the device keeps a MAC address learned through ARP in the device's ARP cache. The device resets the timer to zero each time the ARP entry is refreshed and removes the entry if the timer reaches the ARP age. **NOTE:** You also can change the ARP age on an individual interface basis. Refer to Table 107 on page 658.	Ten minutes	page 688

TABLE 106 IP global parameters (Continued)

Parameter	Description	Default	See page...
Proxy ARP	An IP mechanism a router can use to answer an ARP request on behalf of a host, by replying with the router's own MAC address instead of the host's.	Disabled	page 689
Static ARP entries	An ARP entry you place in the static ARP table. Static entries do not age out.	No entries	page 690
Time to Live (TTL)	The maximum number of routers (hops) through which a packet can pass before being discarded. Each router decreases a packet's TTL by 1 before forwarding the packet. If decreasing the TTL causes the TTL to be 0, the router drops the packet instead of forwarding it.	64 hops	page 709
Directed broadcast forwarding	A directed broadcast is a packet containing all ones (or in some cases, all zeros) in the host portion of the destination IP address. When a router forwards such a broadcast, it sends a copy of the packet out each of its enabled IP interfaces. **NOTE:** You also can enable or disable this parameter on an individual interface basis. Refer to Table 107 on page 658.	Disabled	page 709
Directed broadcast mode	The packet format the router treats as a directed broadcast. The following formats can be directed broadcast: • All ones in the host portion of the packet's destination address. • All zeroes in the host portion of the packet's destination address.	All ones **NOTE:** If you enable all-zeroes directed broadcasts, all-ones directed broadcasts remain enabled.	page 711
Source-routed packet forwarding	A source-routed packet contains a list of IP addresses through which the packet must pass to reach its destination.	Enabled	page 710
Internet Control Message Protocol (ICMP) messages	The PowerConnect can send the following types of ICMP messages: • Echo messages (ping messages) • Destination Unreachable messages • Redirect messages **NOTE:** You also can enable or disable ICMP Redirect messages on an individual interface basis. Refer to Table 107 on page 658.	Enabled	page 712 page 714

TABLE 106 IP global parameters (Continued)

Parameter	Description	Default	See page...
ICMP Router Discovery Protocol (IRDP)	An IP protocol a router can use to advertise the IP addresses of its router interfaces to directly attached hosts. You can enable or disable the protocol, and change the following protocol parameters: • Forwarding method (broadcast or multicast) • Hold time • Maximum advertisement interval • Minimum advertisement interval • Router preference level **NOTE:** You also can enable or disable IRDP and configure the parameters on an individual interface basis. Refer to Table 107 on page 658.	Disabled	page 738
Maximum BootP relay hops	The maximum number of hops away a BootP server can be located from a router and still be used by the router's clients for network booting.	Four	page 744
Maximum Frame Size	You can set a maximum frame size of all Ethernet frames that are forwarded by the system.		page 681
Domain name for Domain Name Server (DNS) resolver	A domain name you can use in place of an IP address for certain operations such as IP pings, trace routes, and Telnet management connections to the router.	None configured	page 679
DNS default gateway addresses	A list of gateways attached to the router through which clients attached to the router can reach DNSs.	None configured	page 679
IP load sharing	A feature that enables the router to balance traffic to a specific destination across multiple equal-cost paths. Load sharing is based on a combination of destination MAC address, source MAC address, destination IP address, source IP address, and IP protocol. **NOTE:** Load sharing is sometimes called Equal Cost Multi Path (ECMP).	Enabled	page 731
Maximum IP load sharing paths	The maximum number of equal-cost paths across which the PowerConnect is allowed to distribute traffic.	Four	page 731
Origination of default routes	You can enable a router to originate default routes for the following route exchange protocols, on an individual protocol basis: • RIP • OSPF • BGP4	Disabled	page 1029
Default network route	The router uses the default network route if the IP route table does not contain a route to the destination and also does not contain an explicit default route (0.0.0.0 0.0.0.0 or 0.0.0.0/0).	None configured	page 730

TABLE 106 IP global parameters (Continued)

Parameter	Description	Default	See page...
Static route	An IP route you place in the IP route table.	No entries	page 714
Source interface	The IP address the router uses as the source address for Telnet, RADIUS, or TACACS/TACACS+ packets originated by the router. The router can select the source address based on either of the following: • The lowest-numbered IP address on the interface the packet is sent on. • The lowest-numbered IP address on a specific interface. The address is used as the source for all packets of the specified type regardless of interface the packet is sent on.	The lowest-numbered IP address on the interface the packet is sent on.	page 685

IP interface parameters

Table 107 lists the interface-level IP parameters for the PowerConnect, their default values, and where to find configuration information.

TABLE 107 IP interface parameters

Parameter	Description	Default	See page...
IP state	The Internet Protocol, version 4	Enabled **NOTE:** You cannot disable IP.	n/a
IP address	A Layer 3 network interface address The PowerConnect has separate IP addresses on individual interfaces.	None configured	page 674
Encapsulation type	The format of the packets in which the router encapsulates IP datagrams. The encapsulation format can be one of the following: • Ethernet II • SNAP	Ethernet II	page 681
IP Maximum Transmission Unit (MTU)	The maximum length (number of bytes) of an encapsulated IP datagram the router can forward.	1500 for Ethernet II encapsulated packets 1492 for SNAP encapsulated packets	page 681
ARP age	Locally overrides the global setting. Refer to Table 106 on page 655.	Ten minutes	page 688
Directed broadcast forwarding	Locally overrides the global setting. Refer to Table 106 on page 655.	Disabled	page 709
ICMP Router Discovery Protocol (IRDP)	Locally overrides the global IRDP settings. Refer to Table 106 on page 655.	Disabled	page 739
ICMP Redirect messages	Locally overrides the global setting. Refer to Table 106 on page 655.	Enabled	page 714

TABLE 107 IP interface parameters (Continued)

Parameter	Description	Default	See page...
DHCP gateway stamp	The router can assist DHCP/BootP Discovery packets from one subnet to reach DHCP/BootP servers on a different subnet by placing the IP address of the router interface that receives the request in the request packet's Gateway field. You can override the default and specify the IP address to use for the Gateway field in the packets. **NOTE:** UDP broadcast forwarding for client DHCP/BootP requests (bootpc) must be enabled and you must configure an IP helper address (the server's IP address or a directed broadcast to the server's subnet) on the port connected to the client.	The lowest-numbered IP address on the interface that receives the request	page 744
UDP broadcast forwarding	The router can forward UDP broadcast packets for UDP applications such as BootP. By forwarding the UDP broadcasts, the router enables clients on one subnet to find servers attached to other subnets. **NOTE:** To completely enable a client's UDP application request to find a server on another subnet, you must configure an IP helper address consisting of the server's IP address or the directed broadcast address for the subnet that contains the server. Refer to the next row.	The router helps forward broadcasts for the following UDP application protocols: • bootps • dns • netbios-dgm • netbios-ns • tacacs • tftp • time	page 741
IP helper address	The IP address of a UDP application server (such as a BootP or DHCP server) or a directed broadcast address. IP helper addresses allow the router to forward requests for certain UDP applications from a client on one subnet to a server on another subnet.	None configured	page 742

GRE IP tunnel

Multi-Service Ironware software supports the tunneling of packets with the Generic Routing Encapsulation (GRE) mechanism over an IP network, as described in RFC 2784. With GRE, packets are encapsulated in a transport protocol packet at a tunnel source and delivered to a tunnel destination, where they are unpacked and made available for delivery.

Considerations in implementing this feature

The considerations in implementing this feature are as follows:

- As a point-to-point tunnel configuration, GRE requires both ends of the tunnel to be configured.
- Only four-byte GRE headers are supported at the ingress (even though eight-byte headers can be processed at a transit node or the egress point).
- A PowerConnect router does not support the key and sequence numbering option with GRE (per RFC 2890).
- The current maximum number of tunnels is 8192 (with default as 256 tunnels).

Figure 119 describes the GRE header format.

FIGURE 119 GRE header format

Checksum – This field is assumed to be zero in this version. If set to 1 means that the **Checksum** (optional) and **Reserved** (optional) fields are present and the Checksum (optional) field contains valid information.

Reserved0 – Bits 6:0 of the field are reserved for future use and must be set to 0 in transmitted packets. If bits 11:7 of the field are non-0, then a receiver must discard the packet. This field is assumed to be 0 in this version.

Ver – This field must be set to 0. This field is assumed to be 0 in this version.

Protocol Type – This field contains the EtherType of the payload protocol.

For details on configuring a GRE IP tunnel, refer to "To display the currently configured Martian addresses refer to "Displaying martian addressing information" on page 774." on page 746.

GRE MTU enhancements

Enhancements have been introduced to support GRE MTU in support of RFC 4459. This includes support for the following:

- Signaling the Lower MTU to the Sources as described in Section 3.2 of RFC 4459
- Fragmentation of the Inner packet as described in Section 3.4 of RFC 4459

This enhancement also allows you to set a specific MTU value for packets entering a configured GRE tunnel. Packets whose size is greater than the configured value are fragmented and encapsulated with IP/GRE headers for transit through the tunnel. This feature supports Jumbo packets although they may be fragmented based on the MTU value configured.

Configuring a GRE IP Tunnel

To configure a GRE IP Tunnel, configure the following parameters:

- "CAM restrictions"
- Maximum Number of Tunnels (optional)
- Tunnel Interface
- Source Address or Source Interface for the Tunnel
- Destination address for the Tunnel
- GRE Encapsulation
- IP address for the Tunnel
- Keep Alive Support (optional)
- TTL Value (optional)
- TOS Value (optional)
- MTU Value (optional)

Configuration considerations

1. To enable keepalive when a GRE source and destination are directly connected, you must disable ICMP redirect on the tunnel source port on the GRE nodes. Otherwise, the keepalive packets go to the CPU where they can degrade CPU performance.
2. Whenever multiple IP addresses are configured on a tunnel source, the primary address of the tunnel is always used for forming the tunnel connections. Consequently, you must carefully check the configurations when configuring the tunnel destination.
3. GRE tunneling is not supported for non-default VRFs.
4. When a GRE tunnel is configured, you cannot configure the same routing protocol on the tunnel through which you learn the route to the tunnel destination. For example, if the PowerConnect learns the tunnel destination route through OSPF protocol, you cannot configure the OSPF protocol on the same Tunnel and vice-versa. When a tunnel has OSPF configured, the PowerConnect cannot learn the tunnel destination route through OSPF. This could cause the system to become unstable.

NOTE
With GRE Dynamic-cam mode, at the Egress node, when a GRE packet is received, Dell programs the CAM entries to forward the packets based on Inner DPA. These host CAM entries will be aging even if the traffic is hitting that CAM entries. This will cause the CAM entries to become aged out and recreated which could cause a small packet loss.

Configuring ECMP for routes through an IP GRE tunnel

If multiple routes are using IP GRE tunnels to a destination, packets are automatically load-balanced between tunnels. This feature allows for load distribution of traffic among the available IP GRE tunnels. If the routes to a destination are both normal IP routes and routes through IP GRE tunnels, ECMP is not enabled.

CAM restrictions

CAMs are partitioned on a PowerConnect router by a variety of profiles that you can select for your specific application.

To implement a CAM partition for a GRE tunnel, enter a command such as the following.

```
NetIron(config)# cam-partition profile ipv4
```

Syntax: [no] cam-partition profile [ipv4 | ipv4-ipv6 | ipv4-vpls | ipv4-vpn | ipv6 | l2-metro | l2-metro-2 | mpls-l3vpn | mpls-l3vpn-2 | mpls-vpls | mpls-vpls-2 | mpls-vpn-vpls | multi-service]

The **ipv4** parameter adjusts the CAM partitions, as described in the tables below to optimize the router for IPv4 applications.

The **ipv4-ipv6** parameter adjusts the CAM partitions, as described in the tables below to optimize the router for IPv4 and IPv6 dual stack applications

The **ipv4-vpls** parameter adjusts the CAM partitions, as described in the tables below to optimize the router for IPv4 and MPLS VPLS applications

The **ipv4-vpn** parameter adjusts the CAM partitions, as described in the tables below to optimize the router for IPv4 and MPLS Layer-3 VPN applications

The **ipv6** parameter adjusts the CAM partitions, as described in the tables below to optimize the router for IPv6 applications.

The **l2-metro** parameter adjusts the CAM partitions, as described in the tables below to optimize the router for Layer 2 Metro applications.

The **l2-metro-2** parameter provides another alternative to **l2-metro** to optimize the router for Layer 2 Metro applications.

The **mpls-l3vpn** parameter adjusts the CAM partitions, as described in the tables below, to optimize the router for Layer 3, BGP or MPLS VPN applications.

The **mpls-l3vpn-2** parameter provides another alternative to **mpls-l3vpn** to optimize the router for Layer 3, BGP or MPLS VPN applications.

The **mpls-vpls** parameter adjusts the CAM partitions, as described in the tables below to optimize the router for MPLS VPLS applications.

The **mpls-vpls-2** parameter provides another alternative to **mpls-vpls** to optimize the router for MPLS VPLS applications. It adjusts the CAM partitions, as described in the tables below.

The **mpls-vpn-vpls** parameter adjusts the CAM partitions, as described in the tables below, to optimize the router for MPLS Layer-3 and Layer-2 VPN applications.

The **multi-service** parameter adjusts the CAM partitions, as described in the tables below to optimize the router for Multi-Service applications.

NOTE
You must reload your PowerConnect router for this command to take effect.

Configuring the maximum number of tunnels supported

You can configure the devices to support a specified number of tunnels using the following command.

```
NetIron(config)# system-max ip-tunnels 512
NetIron(config)# write memory
```

Syntax: system-max ip-tunnels <*number*>

The <*number*> variable specifies the number of GRE tunnels that can be supported.

NOTE
Multicast over GRE tunnels for PIM can support up to the default system max of 256 tunnels if the required hardware resources are available.

NOTE
You must write this command to memory and perform a system reload for this command to take effect.

Configuring a tunnel interface

To configure a tunnel interface, use a the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)
```

Syntax: [no] interface tunnel <*tunnel id*>

The <*tunnel-id*> variable is numerical value that identifies the tunnel being configured. Possible range is from 1 to the maximum configured tunnels in the system.

Configuring a source address or source interface for a tunnel interface

To configure a source address for a specific tunnel interface, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel source 35.0.8.108
```

To configure a source interface for a specific tunnel interface, enter the following command.

```
NetIron(config)# interface tunnel 100
NetIron(config-tnif-100)tunnel source ethernet 3/1
```

Syntax: [no] tunnel source <*ip-address*> **|** <*port-no*>

You can specify either of the following:

The <*ip-address*> variable is the source IP address being configured for the specified tunnel.
The <*port-no*> variable is the source slot or port of the interface being configured for the specified tunnel. When you configure a source interface, there must be at least one IP address configured on that interface. Otherwise, the interface will not be added to the tunnel configuration and an error message like the following will be displayed: " Error - Tunnel source interface 3/1 has no configured ip address.

It can be a physical or virtual interface (ve).

Configuring a destination address for a tunnel interface

To configure a destination address for a specific tunnel interface, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel destination 131.108.5.2
```

Syntax: [no] tunnel destination <*ip-address*>

The <*ip-address*> variable is destination IP address being configured for the specified tunnel.

NOTE
If GRE is configured with a tunnel destination reachable over LAG ports, load balancing will only work with the following LAG types: server LAG or LACP with server LAG. Traffic cannot be load-balanced across multiple ports of a switch LAG.

Configuring a tunnel interface for GRE encapsulation

To configure a specified tunnel interface for GRE encapsulation, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel mode gre ip
```

Syntax: [no] tunnel mode gre ip

The **gre** parameter specifies that the tunnel will use GRE encapsulation

The **Ip** parameter specifies that the tunnel protocol is IP.

Configuring an IP address for a tunnel interface

To configure an IP address for a specified tunnel interface, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)ip address 10.10.3.1/24
```

Syntax: [no] ip address <*ip-address*>

The <*ip-address*> variable is the IP address being configured for the specified tunnel interface.

Configuring keep alive support

This parameter is optional. It lets the router maintain a tunnel in an up or down state based upon the periodic sending of keep alive packets and the monitoring of responses to the packet. If the packets fail to reach the tunnel's far end more frequently than the configured number of retries, the tunnel is placed in a down state. A keep alive packet is a GRE IP packet with no payload.

To configure the keep alive option, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)keepalive 5 4
```

Syntax: [no] keepalive <*seconds*> <*retries*>

The <*seconds*> variable specifies the number of seconds between each initiation of a keep alive message. The range for this interval is 1 – 32767 seconds. The default value is 10 seconds.

The <*retries*> variable specifies the number of times that a packet is sent before the system places the tunnel in the down state. Possible values are from 1 – 255. The default number of retires is 3.

Configuring a TTL value

This is an optional parameter that allows you to set the Time-to-Live value for the outer IP header of the GRE tunnel packets.

To configure the TTL value, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel ttl 100
```

Syntax: [no] tunnel ttl <*ttl-value*>

The <*ttl–value*> variable specifies a TTL value for the outer IP header. Possible values are 1 - 255. The default value is 255.

Configuring a TOS value

This is an optional parameter that allows you to set the TOS value for the outer IP header of the GRE tunnel packets.

To configure the TOS value, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel tos 100
```

Syntax: [no] tunnel tos <*tos-value*>

The <*tos-value*> variable specifies a TOS value for the outer IP header.

Configuring GRE session enforce check

The **gre-session-enforce-check** command lets you enable the GRE session enforce check. When a GRE packet arrives and this feature is enabled, the system tries to match the GRE packet source and destination address pair with the tunnel configured destination and source pair. If the pairs do not match, hardware drops the packet.

To configure the GRE session enforce check, go to the IP tunnel policy context, and then enter the **gre-session-enforce-check** command.

```
NetIron(config)#ip-tunnel-policy
NetIron(config-ip-tunnel-policy)#gre-session-enforce-check
```

Syntax: [no] gre-session-enforce-check

To disable the GRE session enforce check, use the **no** form of this command. This command is disabled by default. You might have to write the configuration to memory and reload the system whenever the configuration of this command is changed because a one-time creation of a source-ingress CAM partition is necessary. The system prompts you if the memory write and reload are required.

The first-time execution of certain commands necessitates the creation of a source-ingress CAM partition, after which you write to memory and reload. These commands are **gre-session-enforce-check**, **ipv6-session-enforce-check**, and **accounting-enable**. After this CAM partition is created, it is not necessary to follow either of the other two commands with a memory write and reload.

Configuring a maximum MTU value for a tunnel interface

You can set an MTU value for packets entering the tunnel. Packets that exceed either the default MTU value of 1476 bytes or the value that you set using this command are fragmented for transit through the tunnel. The default MTU value is set to 1476.

The following command allows you to change the MTU value for packets transiting “tunnel 1”.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel mtu 1500
```

Syntax: [no] tunnel mtu <*packet-size*>

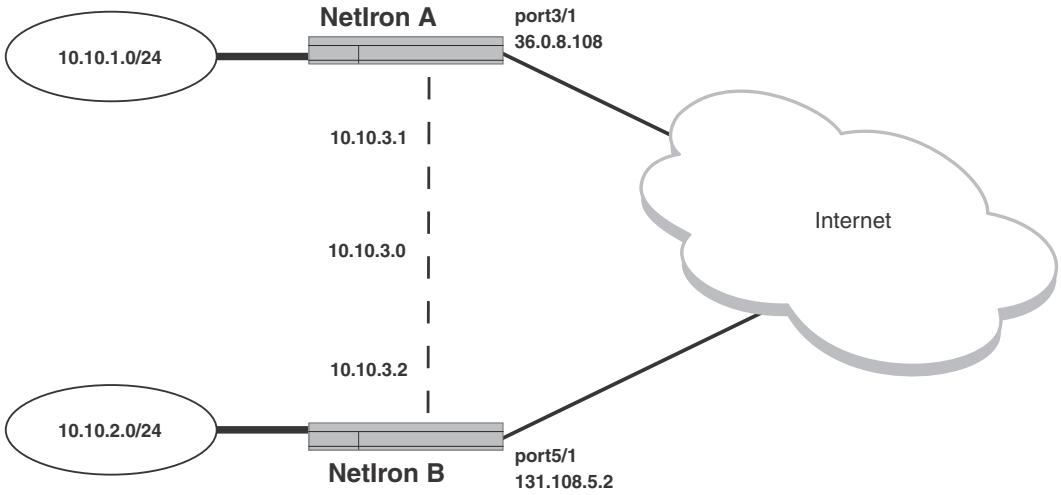
The <*packet-size*> variable specifies the maximum MTU size in bytes for the packets transiting the tunnel.

NOTE
To prevent packet loss after the 24 byte GRE header is added, make sure that any physical interface that is carrying GRE tunnel traffic has an IP MTU setting at least 24 bytes greater than the tunnel MTU setting.

Example of a GRE IP tunnel configuration

In this example, a GRE IP Tunnel is configured between the PowerConnect A router and the PowerConnect B router. Traffic between networks 10.10.1.0/24 and 10.10.2.0/24 is encapsulated in a GRE IP packet sent through the tunnel on the 10.10.3.0 network. and unpacked and sent the destination network. A static route is configured at each router to go through the tunnel interface to the target network.

FIGURE 120 GRE IP tunnel configuration example



Configuration example for PowerConnect A

```
NetIron(config)# interface ethernet 3/1
NetIron(config-int-e10000-3/1)# ip address 36.0.8.108/24
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)# tunnel source 36.0.8.108
NetIron(config-tnif-1)# tunnel destination 131.108.5.2
NetIron(config-tnif-1)# tunnel mode gre ip
NetIron(config-tnif-1)# ip address 10.10.3.1/24
NetIron(config-tnif-1)# keepalive 5 4
NetIron(config-tnif-1)# exit
NetIron(config)# ip route 10.10.2.0/24 10.10.3.2
```

Configuration example for PowerConnect B

```
NetIron(config)# interface ethernet 5/1
NetIron(config-if-e10000-5/1)# ip address 131.108.5.2/24
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)# tunnel source ethernet 5/1
NetIron(config-tnif-1)# tunnel destination 36.0.8.108
NetIron(config-tnif-1)# tunnel mode gre ip
NetIron(config-tnif-1)# keepalive 5 4
NetIron(config-tnif-1)# ip address 10.10.3.2/24
NetIron(config-tnif-1)# exit
NetIron(config)# ip route 10.10.1.0/24 10.10.3.1
```

Displaying GRE tunneling information

You can display GRE tunneling information using the **show ip interface, show ip route** and **show interface tunnel** commands as shown in the following.

```
NetIron# show ip interface tunnel 1
Interface Tunnel 1
  port enabled
  port state: UP
  ip address: 110.255.255.13/24
  Port belongs to VRF: default
  encapsulation: ETHERNET, mtu: 1476
  directed-broadcast-forwarding: disabled
  ip icmp redirect: enabled
  No inbound ip access-list is set
  No outbound ip access-list is set
  No Helper Addresses are configured.
```

Syntax: show ip interface tunnel <*tunnel-no*>

The show ip route command displays routes that are pointing to a GRE tunnel as shown in the following.

```
NetIron# show ip route
     Total number of IP routes: 6
     Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost -
     Destination     Gateway         Port         Cost       Type
     2 10.10.10.0/24  DIRECT          gre_tnl 1     0/0        D

NetIron# show interface tunnel 1
    Tunnel1 is up, line protocol is up
    Hardware is Tunnel
    Tunnel source 45.45.3.3
    Tunnel destination is 45.45.48.1
    Tunnel mode gre ip
     No port name
     Internet address is 110.255.255.13/24,
    Tunnel TOS 0, Tunnel TTL 255 MTU 1476 bytes
    Keepalive is not Enabled
```

Syntax: show interface tunnel <*tunnel-no*>

Multicast over GRE tunnel

Multi-Service Ironware software supports Multicast over a point-to-point GRE tunnel. Multicast over a GRE tunnel allows multicast packets to be transported through a GRE tunnel across an IP cloud towards its receiver. A GRE tunnel is provisioned at each end of the IP cloud. A GRE tunnel is a virtual IP tunnel; the IP tunnel source can also be a VE interface. The IP cloud sitting in between the two GRE endpoints serves as a PIM enabled logical link. As bidirectional control messages are sent over the GRE tunnel, the multicast distribution tree is established across the IP cloud. Multicast data is encapsulated with a predefined GRE header at the ingress node. The GRE packet is routed within the IP cloud using the outer unicast GRE destination address. As the packet reaches the egress node of the tunnel, the packet is decapsulated. The multicast packet continues on its way to the multicast distribution tree to reach its receivers.

Configuring PIM GRE tunnel

The PowerConnect PIM GRE tunnel configuration allows you to enable PIM Sparse (PIM-SM) and PIM Dense (PIM-DM) on a GRE tunnel.

Enabling PIM-SM on a GRE tunnel interface

To enable PIM-SM on a GRE Tunnel Interface, enter the following command.

```
NetIron(config)#interface tunnel 20
NetIron(config-tnif-20)#ip pim-sparse
```

Syntax: [no] ip pim-sparse

Enabling PIM-DM on a GRE tunnel interface

To enable PIM-DM on a GRE Tunnel Interface, enter the following command.

```
NetIron(config)#interface tunnel 20
NetIron(config-tnif-20)#ip pim
```

Syntax: [no] ip pim

Configuring PIM GRE tunnel using the strict RPF check

The NetIron PIM GRE tunnel configuration allows you to enforce strict rpf check rules on (s,g) entry on a GRE tunnel interface. The (s,g) entry uses the GRE tunnel as a RPF interface. During unicast routing transit, GRE tunnel packets may arrive at different physical interfaces. The **ip pim tunnel rpf-strict** command allows you to limit a specific port to accept the (s,g) GRE tunnel traffic.

NOTE
The configuration is not recommended for all users, it is only needed if the user wants to override the default behavior.

When the GRE encapsulated multicast packet is received, hardware processing attempts to find a match in the CAM session based on the inner (s,g) entry. If hardware processing cannot find the inner (s,g) entry in the CAM session, the packet will be dropped. If the **ip pim tunnel rpf-strict** command is configured on a GRE tunnel interface, hardware processing will check on the (s,g) entry, and verify that the packet matches the physical port on the GRE tunnel interface, and the GRE tunnel vlan id.

To limit a specific port to accept the (s,g) GRE tunnel traffic, enter the following command.

```
NetIron(config)#interface tunnel 20
NetIron(config-tnif-20)#ip pim tunnel rpf-strict
```

Syntax: [no] ip pim tunnel [rpf-strict]

The rpf-strict option allows you to set the strict rpf check on the multicast entry.

Tunnel statistics for a GRE tunnel or IPv6 manual tunnel

At a global level, you can enable the collection of statistics for generic routing encapsulation (GRE) tunnels and manual IPv6 tunnels. With this feature, the PowerConnect router collects the statistics for GRE and IPv6 manual tunnels and displays packet counters for tunnels at the management processor (MP). This feature collects and displays unicast and multicast packets over both directions of the tunnels.

Statistics collection is not enabled by default, so you need to enter the IP tunnel policy configuration level and then execute the accounting-enable command to start collecting the statistics for GRE and IPv6 manual tunnels. This procedure is described in "Enabling tunnel statistics" on page 671. This required preliminary ensures that the source-ingress CAM partition is

not allocated unless statistics collection or tunnel session enforcement checks are actually needed. (Because the statistics enable does not enforce the GRE and IPv6 tunnel session checks by default, these capabilities have their own enable commands in the IP tunnel policy CLI level. The applicable commands are described in "Configuring IPv6 session enforce check" on page 672 and "Configuring IPv6 session enforce check" on page 672.) You can view examples of related show command output in "Displaying GRE and manual IPv6 tunnel statistics" on page 770.

The remainder of this introduction to tunnel statistics describes reload behavior for certain commands and detailed notes and restrictions that apply to the support for tunnel statistics.

Reload behavior and the source-ingress CAM partition

When one of the three tunnel-related commands is configured at the CLI level for IP tunnel policy (entered by use of the **ip-tunnel-policy** command), you might need to save the configuration and reload the router to create the required source-ingress-CAM partition. If the memory write and reload are needed, the system prompts for these steps after you finish the enable commands. The condition for which you might need to write and reload is the absence of the source-ingress-CAM partition. If this partition does not exist, the first time that you run either the **gre-session-enforce-check**, **ipv6-session-enforce-check**, or **accounting-enable** command, the system prompts you. Thereafter, when you run any of these three commands to disable or enable a feature, the system does not prompt. Removing any of the configurations can be done at anytime and does not necessitate a reload. The new configuration immediately becomes effective, but the source-ingress CAM partition is removed only upon the next reload.

Operational notes

The subsections that following describe operational characters that relate to the statistics collection.

Source-ingress-CAM partition

The CAM profile restrictions for this feature are the same as those for the tunnel session enforce-check configuration. This feature is not supported in those CAM profiles for which the system cannot allocate the source-ingress-CAM partition that is needed to support the accounting and session check enforcement. The CLI engine checks for compliance and rejects an attempt to enable statistics in this situation. Currently the following CAM profiles are not supported for IP tunnel statistics:

- IPv6
- L2-metro-2
- MPLS-L3VPN-2
- MPLS-VPLS-2
- MPLS-VPN-VPLS

6to4 automatic tunnels

Statistics collection is supported only for manual IPv6 and GRE tunnels. The system does not support statistics collection for 6to4 automatic IPv6 tunnels because, for automatic 6to4 tunnels, only tunnel source-ip is configured, and the destination is known only at runtime when a remote node tries to use this tunnel. The destination points can come up or go down without the local router having any information on how many destinations are to be used for 6to4 tunnels. This uncertainty can cause scalability issues, so neither statistics collection nor session-enforce check are not supported for 6to4 automatic tunnels.

Multicast-over-GRE packets

This feature counts multicast-over-GRE packets. You can see the multicast packet count by using the show interface tunnel <*tunnel-id*> command. You can use other CLI commands to display the aggregate unicast and multicast statistics for the GRE tunnels. For a description of all the applicable show commands, refer to "Displaying GRE and manual IPv6 tunnel statistics" on page 770.

Statistics polling on the MP and LP

The LP module polls the statistics once every second. For every one second, the module polls the statistics either for 5000 entries or until the completion of a specific application. (The same polling mechanism is also used for other applications, such as IP, MPLS, L3VPN, VLL, VPLS and IP Tunnel.) After all the applications are polled, the system waits for 220 seconds to schedule the next polling event. However, the LP module synchronizes statistics to the MP every 30 seconds, so 30 seconds is the granularity of statistics.

The LP synchronizes statistics to the MP in background every 30 seconds, and the MP stores the statistics for all tunnels for every LP module. If a LP module at either the tunnel ingress or egress, the system uses the current stored statistics for that LP module for display (and continue to poll the rest of working modules to get the latest statistics). This mechanism ensures that the tunnel counters never go down (if no clear statistics command is performed on the tunnel).

When a tunnel is down, the LP does not poll the statistics for that tunnel. The LP keeps the old counters as is until you explicitly clear them on the CLI. These counters are displayed when the tunnel is down. When the tunnel comes back up, it resumes polling and adds the new packet counts to the stored statistics and displays the updated statistics.

Clearing the statistics

When you execute the clear statistics tunnel [tunnel-id] command, the operation clears statistics for either one or all of the tunnels regardless of the circumstance—whether the tunnel is up or down, on an ingress or egress module, and so on. Refer to "Clearing GRE tunnel and manual IPv6 tunnel statistics" on page 672 for a description of the clear statistics tunnel command.

Tunneled packets that encounter an ACL

If a packet reaches the ACL permit or deny clauses for the inner IPv4 or IPv6 addresses when it comes through the IP tunnel at the egress node, the packet is not counted as a receive-from-tunnel packet. Instead, it is counted as an ACL packet. You can view ACL packets by using the show access-list accounting command.

Switchover behavior

The LP sends statistics to both the active and the standby MP modules. If an MP switches over, the new-active MP polls the statistics again so it can display the latest statistics. The counters are equal to or greater than the statistics before the switchover for the working modules. If any module goes down before the switchover, the new active MP uses the stored counters to display the statistics for that module.

Hitless operating system upgrade behavior

When a hitless operating system (OS) upgrade occurs, the tunnel statistics are saved and retrieved after the reset of the LP is complete. The system can retrieve the old statistics and do the polling to get the latest PRAM statistics. After the hitless upgrade, the system can display the correct packet counters.

Behavior after an LP failure

If LP module goes down, the counters for that LP are preserved. After the LP comes back up, the preserved counters for that LP can be displayed.

Feature scalability

The system supports statistics for all tunnels because the source ingress CAM partition has 16000 entries that can support the statistics for all tunnels.

Enabling IP tunnel or manual IPv6 statistics

This section describes how to enable and clear statistics for GRE or manual IPv6 tunnels. The enable for this feature is global in scope. The enabling command is one of three enable commands that you run in the IP tunnel policy context of the CLI. (These commands are **gre-session-enforce-check**, **ipv6-session-enforce-check**, and **accounting-enable**. The **ip-tunnel-policy** command puts the CLI in the mode for executing them.) To see examples of tunnel statistics, refer to "Displaying GRE and manual IPv6 tunnel statistics" on page 770.

Enabling tunnel statistics

To enable the GRE tunnel or manual IPv6 tunnel statistics, go to the IP tunnel policy mode of the CLI and execute the accounting-enable command, as the following example illustrates.

```
NetIron(config)#ip-tunnel-policy
NetIron(config-ip-tunnel-policy)#accounting-enable
```

Syntax: [no] accounting-enable

To turn off tunnel statistics gathering, prepend the no keyword to the accounting-enable command.

The system might prompt you to write the configuration to memory and reload the system. If the system has not yet allocated a source-ingress CAM partition, it prompts you to write the results of the current configuration to memory and reload the system.

The first-time execution of certain commands can prompt the allocation of a source-ingress CAM partition that is required by certain features. These commands are **gre-session-enforce-check**, **ipv6-session-enforce-check**, and **accounting-enable**. After this CAM partition is allocated, you do not need to do the memory write and reload after the first-time execution of the other two commands.

Clearing GRE tunnel and manual IPv6 tunnel statistics

You can clear all of the statistics for either one or all tunnels by using the clear statistics tunnel command, as the following example illustrates.

```
NetIron#clear statistics tunnel 1
```

Syntax: clear statistics tunnel [tunnel ID]

To clear statistics for a specific tunnel, include the ID of that tunnel.

Configuring IPv6 session enforce check

You can enable the IPv6 session enforce check by using the **ipv6-session-enforce-check** command. When an IPv6 packet arrives and this feature is enabled, the system tries to match the Ipv6 packet source and destination address pair with the tunnel configured destination and source pair. If the pairs do not match, the packet is dropped in hardware.

To configure the IPv6 session enforce check, go to the IP tunnel policy context and enter the **ipv6-sessionenforce-check** command.

```
NetIron(config)#ip-tunnel-policy
NetIron(config-ip-tunnel-policy)#ipv6-session-enforce-check
```

Syntax: [no] ipv6-session-enforce-check

To disable the IPv6 session enforce check, use the no form of this command.

The system might prompt you to write the configuration to memory and reload the system. If the system has not yet allocated a source-ingress CAM partition, it prompts you to write the results of the current configuration to memory and reload the system.

The first-time execution of certain commands can prompt the allocation of a source-ingress CAM partition that is required by certain features. These commands are gre-session-enforce-check, ipv6-session-enforce-check, and accounting-enable. After this CAM partition is allocated, you do not need to do the memory write and reload after the first-time execution of the other two commands.

NOTE
The **ipv6-sessions-enforce-check** command is not supported for 6to4 automatic tunnels.

Restart global timers

Restart contains two global timers that:

- Limit the amount of time used for re-syncing routes between the backup Management module and Interface modules (LPs) within the same chassis
- Allow a buffer time for protocols to converge and solve dependencies among each other

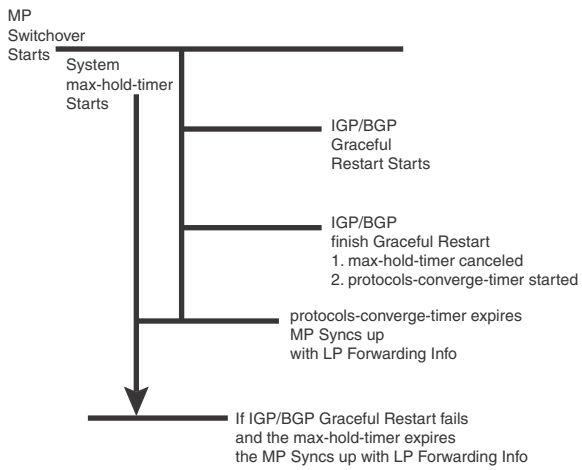
If the protocol-based restart features are configured when a Management module (MP) performs a switchover to the its backup, routes are maintained on the LPs through the protocol-based restart processes for a specified period of time while the new MP learns the network routes. Once the MP learns all of its routes, the routes from the MP are synced with the routes on the LPs.

The two timers introduced here are called the **max-hold-timer** and the **protocols-converge-timer**.

The process of syncing routes between a new MP and its LPs using the new timers are illustrated in Figure 121 and described in the following steps.

1. The MP switchover from active to redundant MP begins.
2. The system **max-hold- timer** starts.
3. The IGP/BGP restart process begins.
4. If the IGP/BGP restart process is completed before the system **max-hold-timer** expires, the system **max-hold-timer** is cancelled and the **protocols-converge-timer** starts.
5. Once the **protocols-converge-timer** expires, the MP syncs up forwarding information with the LPs.
6. If the system **max-hold-timer** expires before the IGP/BGP restart process is completed, the MP syncs up forwarding information with the LPs at that time and the **protocols-converge-timer** is never started.

FIGURE 121 MP to LP re-syncing process



Configuring the graceful-restart max-hold-timer

This timer defines the maximum hold time before a management module syncs up new forwarding information to interface modules during the restart process. While the default value of 300 seconds will work in most cases, if a router is loaded with a very large number of routes and OSPF/BGP peering adjacencies you might want to fine-tune your router's performance by increasing this value.

The value of this timer can be set using the command shown in the following.

```
NetIron(config)# graceful-restart max-hold-timer 500
```

Syntax: [no]graceful-restart max-hold-timer <*hold-time*>

The <*hold-time*> variable is the maximum number of seconds that a management routing module waits before it syncs up new forwarding information to the interface modules during a restart. The range for the hold time is 30 – 3600 seconds. The default time is 300 seconds.

Graceful-restart protocols-converge-timer

This timer defines the time that a PowerConnect router waits for restarting protocols to converge at the final step in the restart process. In a heavily loaded system where BGP/OSPF/GRE/Static protocols can have a dependency on each other, their restart procedures may also depend on each other. This timer is to allow protocols to solve inter-dependencies after individual restart processes and before routing modules sync up new forwarding information to interface module. The default value of 5 seconds will work in most cases but if a system is heavily loaded and has protocols that depend on each other, you might want to fine-tune your system by increasing this value.

The value of this timer can be set using the command shown in the following.

```
NetIron(config)# graceful-restart protocols-converge-timer 20
```

Syntax: [no]graceful-restart protocols-converge-timer <*hold-time*>

The <*hold-time*> variable is the maximum hold time in seconds before management routing modules sync up new forwarding information to interface modules during restart. The range of permissible values is 0 to 1200 seconds. The default value is 5 seconds.

Configuring IP parameters

Some parameters can be configured globally while others can be configured on individual interfaces. Some parameters can be configured globally and overridden for individual interfaces.

Configuring IP addresses

You can configure an IP address on the following types of the PowerConnect interfaces:

- Ethernet port
- Virtual routing interface (also called a Virtual Ethernet or “VE”)
- Loopback interface

By default, you can configure up to 24 IP addresses on each interface.

NOTE
After you configure a virtual routing interface on a VLAN, you cannot configure Layer 3 interface parameters on individual ports in the VLAN. Instead, you must configure the parameters on the virtual routing interface itself.
Also, after an IP address is configured on an interface, the hardware is programmed to route all IP packets that are received on the interface. Consequently, all IP packets not destined for this device’s MAC address are not bridged and are dropped.

The PowerConnect supports both classical IP network masks (Class A, B, and C subnet masks, and so on) and Classless Interdomain Routing (CIDR) network prefix masks.

- To enter a classical network mask, enter the mask in IP address format. For example, enter "209.157.22.99 255.255.255.0" for an IP address with a Class-C subnet mask.
- To enter a prefix network mask, enter a forward slash (/) and the number of bits in the mask immediately after the IP address. For example, enter "209.157.22.99/24" for an IP address that has a network mask with 24 significant bits (ones).

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the display to prefix format.

Assigning an IP address to an Ethernet port

To assign an IP address to port 1/1, enter the following commands.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# ip address 192.45.6.1 255.255.255.0
```

NOTE
You also can enter the IP address and mask in CIDR format, as follows.

```
NetIron(config-if-e10000-1/1)# ip address 192.45.6.1/24
```

Syntax: [no]**interface ethernet** <*slot/port*>

Syntax: [no] **ip address** <*ip-addr*> <*ip-mask*> | <*ip-addr*>/<*mask-bits*> **[ospf-ignore | ospf-passive | secondary]**

The **ospf-ignore** | **ospf-passive** parameters modify the PowerConnect defaults for adjacency formation and interface advertisement. Use one of these parameters if you are configuring multiple IP subnet addresses on the interface but you want to prevent OSPF from running on some of the subnets:

- **ospf-passive** – disables adjacency formation with OSPF neighbors (but does not disable advertisement of the interface into OSPF). By default, when OSPF is enabled on an interface, the software forms OSPF router adjacencies between each primary IP address on the interface and the OSPF neighbor attached to the interface.
- **ospf-ignore** – disables OSPF adjacency formation and advertisement of the interface into OSPF. The subnet is completely ignored by OSPF.

Use the **secondary** parameter if you have already configured an IP address within the same subnet on the interface.

NOTE
When you configure more than one address in the same subnet, all but the first address are secondary addresses and do not form OSPF adjacencies.

Assigning an IP address to a loopback interface

Loopback interfaces are always up, regardless of the states of physical interfaces. They can add stability to the network because they are not subject to route flap problems that can occur due to unstable links between a PowerConnect and other devices.

You can configure up to 64 loopback interfaces on a PowerConnect router.

You can add up to 24 IP addresses to each loopback interface.

NOTE
If you configure the PowerConnect to use a loopback interface to communicate with a BGP4 neighbor, you also must configure a loopback interface on the neighbor and configure the neighbor to use that loopback interface to communicate with the PowerConnect.

To add a loopback interface, entehr commands such as those shown in the following example.

```
NetIron(config-bgp-router)# exit
NetIron(config)# int loopback 1
NetIron(config-lbif-1)# ip address 10.0.0.1/24
```

Syntax: [no]interface loopback <*num*>

For the syntax of the IP address, refer to "Assigning an IP address to an Ethernet port" on page 675.

Assigning an IP address to a virtual interface

A virtual interface is a logical port associated with a Layer 3 Virtual LAN (VLAN) configured on a PowerConnect.

NOTE
Other sections in this chapter that describe how to configure interface parameters also apply to virtual interfaces.

NOTE
The PowerConnect uses the lowest MAC address on the device (the MAC address of port 1 or 1/1) as the MAC address for all ports within all virtual interfaces you configure on the device.

To add a virtual interface to a VLAN and configure an IP address on the interface, enter commands such as the following.

```
NetIron(config)# vlan 2 name IP-Subnet_1.1.2.0/24
NetIron(config-vlan-2)# untag e1/1 to 1/4
NetIron(config-vlan-2)# router-interface ve1
NetIron(config-vlan-2)# interface ve1
NetIron(config-vif-1)# ip address 1.1.2.1/24
```

The first two commands create a Layer 3 protocol-based VLAN named "IP-Subnet_1.1.2.0/24" and add a range of untagged ports to the VLAN. The **router-interface** command creates virtual interface 1 as the routing interface for the VLAN. The last two commands change to the interface configuration level for the virtual interface and assign an IP address to the interface.

Syntax: [no]router-interface ve <*num*>

Syntax: [no]interface ve <*num*>

The <*num*> parameter specifies the virtual interface number. You can specify from 1 to the maximum number of virtual interfaces supported on the device. To display the maximum number of virtual interfaces supported on the device, enter the **show default values** command. The maximum is listed in the System Parameters section, in the Current column of the virtual-interface row.

For the syntax of the IP address, refer to "Assigning an IP address to an Ethernet port" on page 675.

Assigning a MAC address to a virtual interface

By default, the PowerConnect uses the MAC address of the first port (1 or 1/1) as the MAC address for all virtual routing interfaces configured on the device. You can specify a different MAC address for the virtual routing interfaces. If you specify another MAC address for the virtual routing interfaces, the address applies to all the virtual routing interfaces configured on the device. To specify the MAC address for virtual routing interfaces, enter commands such as the following.

```
NetIron(config)# virtual-interface-mac aaaa.bbbb.cccc
NetIron(config)# write memory
NetIron(config)# end
NetIron# reload
```

Syntax: [no] virtual-interface-mac <*mac-addr*>

Enter the MAC address in the following format: HHHH.HHHH.HHHH

NOTE
You must save the configuration and reload the software to place the change into effect.

Deleting an IP address

To delete an IP address, enter a command such as the following.

```
NetIron(config-if-e1000-1/1)# no ip address 1.1.2.1
```

This command deletes IP address 1.1.2.1. You do not need to enter the subnet mask.

To delete all IP addresses from an interface, enter the following command.

```
NetIron(config-if-e1000-1/1)# no ip address *
```

Syntax: no ip address <*ip-addr*>

Enabling hardware forwarding of IP option packets based on L3 destination

The IP option field in an IP header is variable in length. A packet can have zero or more options and an option can have either of the following forms:

- a single octet of option-type
- an option-type octet, an option-length octet, and option-data octets

The option-type octet consists of the following three fields:

- 1 bit copied flag
- 2 bits option class
- 5 bits option number

By default, IP option packets are sent to the CPU for forwarding. When configured on a physical interface, the **ignore-options** command directs the router to ignore all options in IP option packets that are received at the configured port. These packets are then treated as if there were no options configured and forwarded based on their Layer-3 destination. The **ignore-options** command is configured as shown in the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# ignore-options
```

Syntax: [no] ignore-options

This command only applies to IP option packets in the default VRF.

When the **ignore-options** command is configured on a port, RSVP router alert packets incoming on that port will not be sent to the CPU. Consequently, MPLS should not be configured on a physical port where the **ignore-options** command is configured.

Using the ignore-options command in a LAG configuration

The **ignore-options** command can be used on a LAG but it must apply to all ports on the LAG. This applies to both static and LACP LAGs as described in the following:

Configuring the ignore-options command on a static LAG

To configure the **ignore-options** command on a static LAG, each port on the LAG must be configured with the command. You can do this by configuring the command on each port before the LAG configuration or configuring the **ignore-options** command on the primary port of the LAG which automatically applies the command to all ports on the LAG as shown in the following.

```
NetIron(config)# trunk e 3/1 to 3/4

trunk transaction done.
NetIron(config-trunk-3/1-3/4)# exit
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e1000-3/1)# ignore-options
```

If the LAG is removed, the **ignore-options** command will be propagated to all ports that were previously in the LAG.

If you try to create a LAG where some of the ports have the **ignore-options** command configured and some do not, the LAG will not be allowed as shown in the following example.

```
NetIron(config)# trunk e 3/1 to 3/2
port 3/1 ignore-options is Enabled, but port 3/2 ignore-options is Disabled
Error: port 3/1 and port 3/2 have different configurations

trunk transaction failed:  trunk Config Vetoed
```

Configuring the ignore-options command on a LACP LAG

Just as with static LAGs, if you want to configure the **ignore-options** command on an LACP LAG, the command must be enabled on all ports within the LAG. If it is not, the LACP LAG will not be accepted as shown in the following.

```
NetIron(config)#lag sta_lag static
NetIron(config-lag-sta_lag)#ports e 1/3 to 1/4
NetIron(config-lag-sta_lag)#primary-port 1/3
NetIron(config-lag-sta_lag)#deploy
NetIron(config-lag-sta_lag)#int e 1/3
NetIron(config-if-e1000-1/3)#ignore-options

NetIron(config)#lag sta_lag static
NetIron(config-lag-sta_lag)#ports e 1/3 e 1/4
NetIron(config-lag-sta_lag)#primary-port 1/3
NetIron(config-lag-sta_lag)#deploy
port 1/3 ignore-options is Enabled, but port 1/4 ignore-options is Disabled
Error: port 1/3 and port 1/4 have different configurations
LAG sta_lag deployment failed!

NetIron(config)#int e 1/3
```

```
NetIron(config-if-e1000-1/3)#ignore-options
NetIron(config-if-e1000-1/3)#lag dyn_lag dynamic
NetIron(config-lag-dyn_lag)#ports e 1/3 e 1/4
NetIron(config-lag-dyn_lag)#primary-port 1/3
NetIron(config-lag-dyn_lag)#deploy
port 1/3 ignore-options is Enabled, but port 1/4 ignore-options is Disabled
Error: port 1/3 and port 1/4 have different configurations
LAG dyn_lag deployment failed!
```

Configuring domain name server (DNS) resolver

The DNS resolver lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a PowerConnect and thereby recognize all hosts within that domain. After you define a domain name, the PowerConnect automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain “newyork.com” is defined on a PowerConnect and you want to initiate a ping to host “NYC01” on that domain, you need to reference only the host name in the command instead of the host name and its domain name. For example, you could enter either of the following commands to initiate the ping.

```
NetIron# ping nyc01
NetIron# ping nyc01.newyork.com
```

Multiple DNS queries can be executed simultaneously, making it possible for the PowerConnect device to run multiple simultaneous Telnet, ping or traceroute commands using host names.

Defining an IPv4 DNS entry

You can define up to four DNS servers for each DNS entry. The first entry serves as the primary default address. If a query to the primary address fails to be resolved after three attempts, the next gateway address is queried (also up to three times). This process continues for each defined gateway address until the query is resolved. The order in which the default gateway addresses are polled is the same as the order in which you enter them.

Suppose you want to define the domain name of newyork.com on a device and then define four possible default DNS gateway addresses. To do so using IPv4 addressing, you would enter the following commands.

```
NetIron(config)# ip dns domain-name newyork.com
NetIron(config)# ip dns server-address 209.157.22.199 205.96.7.15 208.95.7.25
201.98.7.15
```

Syntax: [no] ip dns server-address <*ip-addr*> [<*ip-addr*>] [<*ip-addr*>] [<*ip-addr*>]

In this example, the first IP address in the **ip dns server-address...** command becomes the primary gateway address and all others are secondary addresses. Because IP address 201.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

DNS queries of IPv4 and IPv6 DNS servers

IPv4 and IPv6 DNS record queries search through IPv4 and IPv6 DNS servers as described in the following:

For IPv4 DNS record queries:

- Loop thru all configured IPv4 DNS servers,

- If no IPv4 DNS servers were configured, then loop through all configured IPv6 DNS servers (if any).

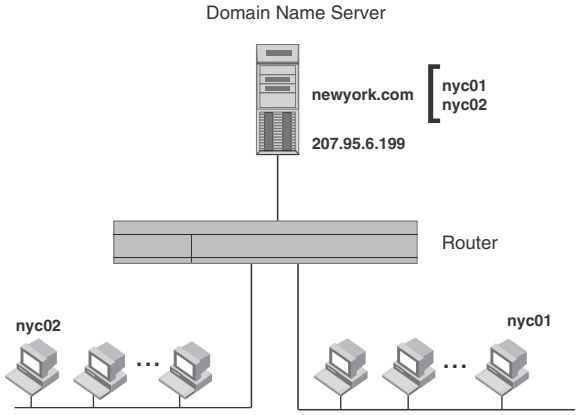
For IPv6 DNS record queries:

- Loop thru all configured IPv6 DNS servers,
- If no IPv6 DNS servers were configured, then loop through all configured IPv4 DNS servers (if any).

Using a DNS name to initiate a trace route

Suppose you want to trace the route from a PowerConnect to a remote server identified as NYC02 on domain newyork.com.

FIGURE 122 Querying a host on the newyork.com domain



Because the newyork.com domain is already defined on the PowerConnect, you need to enter only the host name, NYC02, as noted below.

```
NetIron# traceroute nyc02
```

Syntax: [no] traceroute <*host-ip-addr*> **[maxttl** <*value*>*] [minttl* <*value*>**] [numeric] [timeout** <*value*>**]**
[source-ip <*ip addr*>**]**

The only required parameter is the IP address of the host at the other end of the route.

After you enter the command, a message indicating that the DNS query is in process and the current gateway address (IP address of the domain name server) being queried appear on the screen.

```
Type Control-c to abort
Sending DNS Query to 209.157.22.199
Tracing Route to IP node 209.157.22.80
To ABORT Trace Route, Please use stop-traceroute command.
 Traced route to target IP node 209.157.22.80:
   IP Address          Round Trip Time1    Round Trip Time2
  207.95.6.30         93 msec             121 msec
```

NOTE
In the above example, 209.157.22.199 is the IP address of the domain name server (default DNS gateway address), and 209.157.22.80 represents the IP address of the NYC02 host.

Using Telnet and Secure Shell

Up to six inbound and five outbound Telnet connections can be supported simultaneously by the PowerConnect device. For detailed information about the use of Telnet on the PowerConnect, please see "Using Telnet" on page 2011. The PowerConnect also supports Secure Shell (SSH) access to management functions; for further information, please refer to "Using Secure Shell" on page 2013.

Changing the encapsulation type for IP packets

The PowerConnect encapsulates IP packets into Layer 2 packets, to send the IP packets on the network. A Layer 2 packet is also called a MAC layer packet or an Ethernet frame. The MAC address of the PowerConnect interface sending the packet is the source address of the Layer 2 packet. The Layer 2 packet's destination address can be one of the following:

- The MAC address of the IP packet's destination. In this case, the destination device is directly connected to the PowerConnect.
- The MAC address of the next-hop gateway toward the packet's destination.
- An Ethernet broadcast address.

The entire IP packet, including the source address, destination address, other control information, and the data, is placed in the data portion of the Layer 2 packet. Typically, an Ethernet network uses one of two different formats of Layer 2 packet:

- Ethernet II
- Ethernet SNAP (also called IEEE 802.3)

The control portions of these packets differ slightly. All IP devices on an Ethernet network must use the same format. The PowerConnect uses Ethernet II by default. You can change the IP encapsulation to Ethernet SNAP on individual ports if needed.

NOTE
All devices connected to the PowerConnect port must use the same encapsulation type.

To change the IP encapsulation type on interface 1/5 to Ethernet SNAP, enter the following commands.

```
NetIron(config)# int e 1/5
NetIron(config-if-e1000-1/5)# ip encapsulation snap
```

Syntax: [no] ip encapsulation snap | ethernet-2

Setting the maximum frame size globally

You can set the default maximum frame size to control the maximum size of Ethernet frames that the Ethernet MAC framers will accept or transmit. The size is counted from the beginning of Ethernet header to the end of CRC field. The default maximum frame size must be greater than an IP MTU value set using the "Globally changing the IP MTU" on page 683.

To set a maximum frame size that applies to the router for Ethernet ports, enter a command such as the following.

```
NetIron(config)# default-max-frame-size 2000
NetIron(config)# write memory
NetIron(config)# reload
```

Syntax: [no] default-max-frame-size <*bytes*>

Enter 64 – 9216 for <*bytes*>. If you have POS interface modules installed in a router, you need to consider the value of the pos-default-max-frame-size setting as described in "Setting the maximum frame size globally for POS modules" on page 190.

NOTE
You must run the **write memory** command and reload the PowerConnect router for the **default-max-frame-size** command to take effect.

Changing the MTU

The IP MTU is the maximum length of an IP packet that a Layer 2 packet can contain. If an IP packet is larger than the IP MTU allowed by the Layer 2 packet, the PowerConnect fragments the IP packet into multiple parts that will fit into Layer 2 packets, and sends the parts of the fragmented IP packet separately, in different Layer 2 packets. The device that receives the multiple fragments of the IP packet reassembles the fragments into the original packet. The default IP MTU is 1500 bytes for Ethernet II packets. You can change the IP MTU globally or for individual IP interfaces. You can increase the IP MTU size to accommodate large packet sizes, such as jumbo packets, globally or on individual IP interfaces. However, IP MTU cannot be set higher than the maximum frame size, minus 18.

NOTE
For multicast data traffic, frames are not fragmented and the IP mtu setting is ignored.

For jumbo packets, the PowerConnect supports hardware forwarding of Layer 3 jumbo packets. Layer 3 IP unicast jumbo packets received on a port that supports the frame's IP MTU size and forwarded to another port that also supports the frame's IP MTU size are forwarded in hardware.

NOTE
Policy Based Routing (PBR) currently does not support this IP MTU feature.

Configuration considerations for increasing the IP MTU:

- The maximum value of an IP MTU cannot exceed the configured maximum frame size, minus 18. For example, global IP MTU cannot exceed the value of **default-max-frame-size**, minus 18 bytes. IP MTU for an interface cannot exceed the value of the maximum frame size configured, minus 18 bytes. The 18 bytes is used for IP overhead, VLAN tagging, etc.
- When you increase the IP MTU size of for an IP interface, the increase uses system resources. Increase the IP MTU size only on the IP interfaces that need it. For example, if you have one IP interface connected to a server that uses jumbo frames and two other IP interfaces connected to clients that can support the jumbo frames, increase the IP MTU only on those three IP interfaces. Leave the IP MTU size on the other IP interfaces at the default value (1500 bytes). Globally increase the IP MTU size only if needed.

How To determine the actual MTU value

An IPv4 interface can obtain it's MTU value from any of the following sources:

- Default IP MTU setting
- Global MTU Setting
- Interface MTU Setting

An interface determines its actual MTU value through the process described below.

1. If an IPv4 Interface MTU value is configured, that value will be used.
2. If an IPv4 Interface MTU value is not configured and an IPv4 Global MTU value is configured, the configured global MTU value will be used.
3. If neither an IPv4 Interface MTU value or an IPv4 Global MTU value are configured, the default IPv4 MTU value of 1500 will be used.

Globally changing the IP MTU

At the global CLI level, the **ip mtu** command has been deprecated. The new command is **ip global-mtu** command. After upgrading to release 04.1.00, the old command syntax will no longer be accepted except at system bootup time.

To globally enable jumbo support on all IP interfaces, enter commands such as the following.

```
NetIron(config)# ip global-mtu 5000
NetIron(config)# write memory
```

If the old command syntax is entered, the following error message will display on the console:

```
NetIron(config)#ip mtu 800
ERROR - Command deprecated. Keyword mtu is replaced by global-mtu.
```

Syntax: [no] ip global-mtu <*bytes*>

The <*bytes*> parameter specifies the maximum IP packet size to be forwarded on a port. You may enter any number within the range of 576 – 9198. However, this value must be 18 bytes less than the value of the global maximum frame size.

If you are using POS interface modules in your PowerConnect router, the value must not exceed the value set using the **pos-default-max-frame-size** command.

Changing the maximum transmission unit on an individual interface

By default, the maximum IP MTU sizes are as follows:

- 1500 bytes – The maximum for Ethernet II encapsulation

NOTE
The IP MTU configured at the IP interface level takes precedence over the IP MTU configured at the global level for that IP interface.

To change the IP MTU for interface 1/5 to 1000, enter the following commands.

```
NetIron(config)# int e 1/5
NetIron(config-if-e10000-5)# ip mtu 1000
```

Syntax: [no] ip mtu <bytes>

The <*bytes*> variable specifies the IP MTU. However, the value of IP MTU on an interface cannot exceed the configured value **default-max-frame-size**, minus 18 bytes. The default IP MTU for Ethernet II packets is 1500.

Changing the router ID

In most configurations, a PowerConnect has multiple IP addresses, usually configured on different interfaces. As a result, a PowerConnect's identity to other devices varies depending on the interface to which the other device is attached. Some routing protocols, including OSPF and BGP4, identify a PowerConnect by just one of the IP addresses configured on the PowerConnect, regardless of the interfaces that connect the PowerConnect devices. This IP address is the router ID.

NOTE
RIP does not use the router ID.

NOTE
If you change the router ID, all current BGP4 sessions are cleared.

By default, the router ID on a PowerConnect is one of the following:

- If the router has loopback interfaces, the default router ID is the IP address configured on the lowest numbered loopback interface configured on the PowerConnect. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default router ID is 9.9.9.9/24:
  - Loopback interface 1, 9.9.9.9/24
  - Loopback interface 2, 4.4.4.4/24
  - Loopback interface 3, 1.1.1.1/24
- If the device does not have any loopback interfaces, the default router ID is the lowest numbered IP interface configured on the device.

If you prefer, you can explicitly set the router ID to any valid IP address. The IP address should not be in use on another device in the network.

You can set a router ID for a specific VRF as described within this section. In order to make the route ID calculation more deterministic, the router calculates the router-id value during bootup and does not calculate or change the router-id value unless the IP address used for the router-id value on the router is deleted, or the **clear router-id** command is issued. Additionally, setting a router-id value overrides the existing router-id value and takes effect immediately. Once a router-id value set by a user is removed using the **no ip router-id x.x.x.x** command, the router will again recalculate the router-id value based on current information.

NOTE
The PowerConnect uses the same router ID for both OSPF and BGP4. If the router is already configured for OSPF, you may want to use the router ID that is already in use on the router rather than set a new one. To display the router ID, enter the **show ip** command at any CLI level.

To change the router ID, enter a command such as the following.

```
NetIron(config)# ip router-id 209.157.22.26
```

Syntax: [no]ip router-id <*ip-addr*>

The <*ip-addr*> can be any valid, unique IP address.

To set the router ID within a VRF, enter a command such as the following.

```
NetIron(config)# ip vrf blue
NetIron(config-ip-vrf-blue)# ip router-id 209.157.22.26
```

Syntax: [no]ip router-id <*ip-addr*>

NOTE
The command for setting the router ID for a specified VRF is exactly the same as for the default VRF. The only difference is that when setting it for a specific VRF, the **ip router-id** command is configured within the VRF as shown in the example.

NOTE
You can specify an IP address used for an interface, but do not specify an IP address in use by another device.

Recalculating the router ID

You can use the **clear ip router-id** command to direct a router to recalculate the IP router ID. This can be done for the default VRF or for a specified VRF, as shown in the following.

```
NetIron(config)# clear ip router-id
```

Syntax: **clear ip router-id [vrf** <*vrf-name*> **]**

Using this command without the **vrf** option recalculates the IP router ID for the default VRF.

You can use the **vrf** option to recalculate the IP router ID for a specific VRF that is specified by the <*vrf-name*> variable.

Specifying a single source interface for Telnet, SSH, SNTP, TFTP, TACACS/TACACS+, or RADIUS packets

When the PowerConnect originates a Telnet, SSH, SNTP, TFTP, TACACS/TACACS+, or RADIUS packet, the source address of the packet is the lowest-numbered IP address on the interface that sends the packet. You can configure the PowerConnect to always use the lowest-numbered IP address on a specific interface as the source addresses for these types of packets. When you configure the PowerConnect to use a single source interface for all Telnet, TACACS/TACACS+, or RADIUS packets, the PowerConnect uses the same IP address as the source for all packets of the specified type, regardless of the ports that actually sends the packets.

Identifying a single source IP address for Telnet, SSH, SNTP, TFTP, TACACS/TACACS+, or RADIUS packets provides the following benefits:

- If your Telnet, SSH, SNTP, TFTP, TACACS/TACACS+, or RADIUS server is configured to accept packets only from specific IP addresses, you can use this feature to simplify configuration of the server by configuring the device to always send the packets from the same link or source address.
- If you specify a loopback interface as the single source for Telnet, SSH, SNTP, TFTP, TACACS/TACACS+, or RADIUS packets, servers can receive the packets regardless of the states of individual links. Thus, if a link to the server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, SSH, SNTP, TFTP, TACACS/TACACS+, or RADIUS packets. You can configure a source interface for one or more of these types of packets separately.

Configuring an interface as the source for Syslog packets

You can configure the device to use the lowest-numbered IP or IPv6 address configured on a loopback interface, virtual interface, or Ethernet port as the source for all Syslog packets from the device. The software uses the lowest-numbered IP or IPv6 address configured on the interface as the source IP address for the packets.

For example, to specify the lowest-numbered IP address configured on a virtual interface as the device's source for all Syslog packets, enter commands such as the following.

```
NetIron(config)# int ve 1
NetIron(config-vif-1)# ip address 10.0.0.4/24
NetIron(config-vif-1)# exit
NetIron(config)# ip syslog source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.4/24 to the interface, then designate the interface's address as the source address for all Syslog packets.

Syntax: [no] ip syslog source-interface ethernet [<*slotnum*>**/]**<*portnum*> **| loopback** <*num*> **| ve** <*num*>

The <*num*> parameter is a loopback interface or virtual interface number. If you specify an Ethernet, the <*slotnum*>/]<*portnum*> is the port's number including the slot number, if you are configuring a device.

The default is the lowest-numbered IP or IPv6 address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

With this new command, the source ip of syslog is no longer controlled by the **snmp-server trap-source** command.

Configuring ARP parameters

Address Resolution Protocol (ARP) is a standard IP protocol that enables the PowerConnect to obtain the MAC address of another device's interface when the PowerConnect knows the IP address of the interface. ARP is enabled by default and cannot be disabled.

How ARP works

The PowerConnect router needs to know a destination's MAC address when forwarding traffic, because the PowerConnect router encapsulates the IP packet in a Layer 2 packet (MAC layer packet) and sends the Layer 2 packet to a MAC interface on a device directly attached to the PowerConnect router. The device can be the packet's final destination or the next-hop router toward the destination.

The PowerConnect router encapsulates IP packets in Layer 2 packets regardless of whether the ultimate destination is locally attached or is multiple router hops away. Since the PowerConnect router's IP route table and IP forwarding cache contain IP address information but not MAC address information, the PowerConnect router cannot forward IP packets based solely on the information in the route table or forwarding cache. The PowerConnect router needs to know the MAC address that corresponds with the IP address of either the packet's locally attached destination or the next-hop router that leads to the destination.

For example, to forward a packet whose destination is multiple router hops away, the PowerConnect router must send the packet to the next-hop router toward its destination, or to a default route or default network route if the IP route table does not contain a route to the packet's destination. In each case, the PowerConnect router must encapsulate the packet and address it to the MAC address of a locally attached device, the next-hop router toward the IP packet's destination.

To obtain the MAC address required for forwarding a datagram, the PowerConnect router does the following:

- First, the PowerConnect router looks in the ARP cache (not the static ARP table) for an entry that lists the MAC address for the IP address. The ARP cache maps IP addresses to MAC addresses. The cache also lists the port attached to the device and, if the entry is dynamic, the age of the entry. A dynamic ARP entry enters the cache when the PowerConnect receives an ARP reply or receives an ARP request (which contains the sender's IP address and MAC address). A static entry enters the ARP cache from the static ARP table (which is a separate table) when the interface for the entry comes up.

  To ensure the accuracy of the ARP cache, each dynamic entry has its own age timer. The timer is reset to zero each time the PowerConnect router receives an ARP reply or ARP request containing the IP address and MAC address of the entry. If a dynamic entry reaches its maximum allowable age, the entry times out and the software removes the entry from the table. Static entries do not age out and can be removed only by you.
- If the ARP cache does not contain an entry for the destination IP address, the PowerConnect router broadcasts an ARP request out all its IP interfaces. The ARP request contains the IP address of the destination. If the device with the IP address is directly attached to the PowerConnect router, the device sends an ARP response containing its MAC address. The response is a unicast packet addressed directly to the PowerConnect router. The PowerConnect router places the information from the ARP response into the ARP cache.

  ARP requests contain the IP address and MAC address of the sender, so all devices that receive the request learn the MAC address and IP address of the sender and can update their own ARP caches accordingly.

  NOTE
  The ARP request broadcast is a MAC broadcast, which means the broadcast goes only to devices that are directly attached to the PowerConnect router. A MAC broadcast is not routed to other networks. However, some routers, including the PowerConnect router, can be configured to reply to ARP requests from one network on behalf of devices on another network. Refer to "Enabling proxy ARP" on page 689.

NOTE
If the router receives an ARP request packet that it is unable to deliver to the final destination because of the ARP timeout and no ARP response is received (the PowerConnect router knows of no route to the destination address), the router sends an ICMP Host Unreachable message to the source.

Rate limiting ARP packets

For rate-limiting purposes, ARP traffic destined for the CPU is assigned a separate global QoS ID 0xFFE. You can configure the rate-limit parameters using the following global CONFIG command.

```
NetIron(config)# ip rate-limit arp policy-map <policy-map-name>
```

By default, the rate-limit parameters for QoS ID 0xFFE will be initialized to allow line-rate traffic. The rate-limit parameters specified using the policy-map are applicable on a per-PPCR basis.

To display ARP accounting statistics, enter the following command.

```
NetIron(config)# show rate-limit arp
```

This command displays the byte counters corresponding to QoS ID 0xFFE.

```
NetIron(config)# clear rate-limit arp
```

This command clears the byte counters corresponding to QoS ID 0xFFE.

When priority-based rate limiting is enabled, QoS IDs 0x3FE, 0x7FE and 0xBFE will be re-mapped to 0xFFE. When priority-based rate limiting is disabled, QoS IDs 0x3FE, 0x7FE and 0xBFE will not be re-mapped to 0xFFE. In either case, only QoS ID 0xFFE will be added to the list of used QoS IDs.

To enable the dynamic addition, deletion, or change in rate-limit values of a policy-map, enter the following command.

```
NetIron(config)# ip rate-limit arp policy-map <policy-map-name>
```

This command takes effect automatically, without unbinding and rebinding the ARP RL policy. If the ARP Rate Limit policy specifies an undefined policy-map, rate limit values are initialized to line-rate values. Dynamic enabling and disabling of priority based rate limiting on a global basis takes effect automatically for the ARP RL policy.

NOTE
ARP packets destined for the CPU will be not be rate-limited by interface-level L2 RL-ACLs. To rate-limit switched ARP packets using interface-level L2 ACLs, you must define an explicit ACL filter with an “etype arp” option, as shown in the following example:

To define an explicit ACL filter, enter commands similar to the following.

```
NetIron(config)# access-list 410 permit any any any etype arp
NetIron(config)# int eth 4/1
NetIron(config-if-e10000-4/1)# rate-limit in access-gr 410 policy-map view
```

NOTE
Since ARP packets are broadcast packets, ARP packets are switched by default within a VLAN by the CPU. Thus to rate-limit switched ARP packets using interface-level L2 ACLs, you must also configure vlan-cpu-protection.

Changing the ARP aging period

When the PowerConnect places an entry in the ARP cache, the PowerConnect router also starts an aging timer for the entry. The aging timer ensures that the ARP cache does not retain learned entries that are no longer valid. An entry can become invalid when the device with the MAC address of the entry is no longer on the network.

The ARP age affects dynamic (learned) entries only, not static entries. The default ARP age is ten minutes. On the PowerConnect router, you can change the ARP age to a value from 0 – 240 minutes. If you set the ARP age to zero, aging is disabled and entries do not age out.

To globally change the ARP aging parameter to 20 minutes, enter the following command.

```
NetIron(config)# ip arp-age 20
```

Syntax: [no]ip arp-age <*num*>

The <*num*> parameter specifies the number of minutes and can be from 0 – 240. The default is 10. If you specify 0, aging is disabled.

To override the globally configured IP ARP age on an individual interface, enter a command such as the following at the interface configuration level.

```
NetIron(config-if-e1000-1/1)# ip arp-age 30
```

Enabling proxy ARP

Proxy ARP allows the PowerConnect router to answer ARP requests from devices on one network on behalf of devices in another network. Since ARP requests are MAC-layer broadcasts, they reach only the devices that are directly connected to the sender of the ARP request. Thus, ARP requests do not cross routers.

For example, if Proxy ARP is enabled on the PowerConnect router connected to two subnets, 10.10.10.0/24 and 20.20.20.0/24, the PowerConnect router can respond to an ARP request from 10.10.10.69 for the MAC address of the device with IP address 20.20.20.69. In standard ARP, a request from a device in the 10.10.10.0/24 subnet cannot reach a device in the 20.20.20.0 subnet if the subnets are on different network cables, and thus is not answered.

NOTE
An ARP request from one subnet can reach another subnet when both subnets are on the same physical segment (Ethernet cable), since MAC-layer broadcasts reach all the devices on the segment.

Proxy ARP is disabled by default.

To enable IP proxy ARP, enter the following command.

```
NetIron(config)# ip proxy-arp
```

To again disable IP proxy ARP, enter the following command.

```
NetIron(config)# no ip proxy-arp
```

Syntax: [no] ip proxy-arp

Enabling local proxy ARP

Under some Layer-2 configurations such as uplink-switch or private VLAN, broadcast packets are not flooded to every port in a VLAN. In these configurations, an ARP request from one host may not reach another host. Enabling the Local Proxy ARP feature on a port directs the router to reply on behalf of a target host if it exists. The ARP reply returned contains the router's mac address instead of the mac address of the target host. In this transaction, the traffic sent to the target host is Layer-3 forwarded rather than Layer-2 switched.

To enable Local Proxy ARP, the global-level command **ip proxy-arp** must first be enabled as described in "Enabling proxy ARP" on page 689. After **ip proxy-arp** has been enabled globally, Local Proxy ARP can be enabled on a specified interface using the following command.

```
NetIron(config-if-e1000-1/1)# interface ethernet 1/1
NetIron(config)# ip local-proxy-arp
```

Syntax: [no] ip local-proxy-arp

Disabling gratuitous ARP requests for local proxy ARP

When the Local Proxy ARP is configured under the IP interface, the PowerConnect router will reply to ARP requests on behalf of the hosts inside the subnet using its own MAC address. Refer to "Enabling local proxy ARP" on page 689 for information on configuring the **Local Proxy ARP** command. In this configuration, when a host comes up, the host tries to ping its own IP address to make sure there is no duplicated IP address by issuing a gratuitous ARP request to its only IP address. The PowerConnect router will reply to this request because it is required under the Local Proxy ARP configuration. When the host receives the ARP reply, the host incorrectly assumes that there is another host using the same IP address.

A gratuitous ARP request packet is defined as an ARP request packet with the sender protocol address that equals to the target protocol address. By disabling Gratuitous ARP Requests for Local Proxy ARP, you are able to control whether to reply to gratuitous ARP requests under the Local Proxy ARP configuration.

To enable the ignore-gratuitous-arp parameter when the **ip local-proxy-arp** command is turned on, enter the following command.

```
NetIron(config-if-e1000-1/6)# ip local-proxy-arp ignore-gratuitous-arp
```

To disable only the ignore-gratuitous-arp parameter when the Local Proxy ARP is configured, enter the following command.

```
NetIron(config-if-e1000-1/6)# no ip local-proxy-arp ignore-gratuitous-arp
```

To disable both the **Local Proxy ARP** command and the ignore-gratuitous-arp parameter, enter the following command.

```
NetIron(config-if-e1000-1/6)# no ip-local-proxy-arp
```

Syntax: [no] ip local-proxy-arp [ignore-gratuitous-arp]

When using the **no ip local-proxy-arp ignore-gratuitous-arp** command, only the ignore-gratuitous-arp parameter is turned off. The **ip-local-proxy-arp** command is still turned on.

The PowerConnect drops all ARP packets that are sent from its own interface. When the ignore-gratuitous-arp parameter is turned on, the PowerConnect will not reply to a gratuitous ARP request even if the target protocol address matches the configured interface IP address.

Creating static ARP entries

The PowerConnect router has a static ARP table, in addition to the regular ARP cache. The static ARP table contains entries that you configure.

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the PowerConnect router, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the device receives an ARP request from the device that has the entry's address.

You can increase the number of configurable static ARP entries. Refer to "Changing the ARP timer" on page 691.

To display the ARP cache and static ARP table, refer to the following:

- To display the ARP table, refer to "Displaying the ARP cache" on page 759.
- To display the static ARP table, refer to "Displaying the static ARP table" on page 760.

- To create a static ARP entry for a static MAC entry, refer to "Creating ARP entries" on page 695.

Changing the ARP timer

When an entry is initially added to the ARP table, it is listed as "Pending." When it is in this state, a series of ARP requests are made to determine if it is a valid entry. If the first attempt succeeds, the status of the entry is changed to "dynamic." It is then subject to the normal rules for dynamic entries. If three attempts fail, the entry is removed from the table.

The ARP timer determines the amount of time that elapses after the ARP request is sent before determining that the request has failed.The **arp-timer** command allows you change the length of the ARP timer as shown in the following.

```
NetIron(config)# ip arp timer 12
```

Syntax: [no] ip arp-timer <*timer-value*>

The <*timer-value*> variable has now been changed so that you are able to enter a value between 1 and 500. Each increment represents 100 ms. Consequently, the minimum value of 1 equals 100 ms.

The default value is 10 which equals 1 sec.

This value can be used to adjust how frequently an ARP request is sent out for a pending ARP entry.

Changing the ARP pending retry timer

The ARP Pending Retry Timer for PowerConnect will send out three ARP request packets for the configured period until ARP is resolved to prevent large amounts of ARP requests from flooding the network during network host scanning activity. The ARP Pending Retry Timer is configurable depending upon the requirements of your system configurations.

The **arp-pending-retry-timer** command allows you to change the length of the ARP pending retry timer as shown in the following.

```
NetIron(config)# ip arp-pending-retry-timer 120
```

Syntax: [no] ip arp-pending-retry-timer <*timer-value*>

The <*timer-value*> variable is a value between 10 to 3600 seconds. The default value is 60 seconds.

Dynamic ARP inspection

NOTE
This feature is supported on Layer 2 and Layer 3 code.

Dynamic ARP Inspection (DAI) enables the device to intercept and examine all ARP request and response packets in a subnet and discard those packets with invalid IP to MAC address bindings. DAI can prevent common man-in-the-middle (MiM) attacks such as ARP cache poisoning, and disallow mis-configuration of client IP addresses.

ARP poisoning

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. Before a host can talk to another host, it must map the IP address to a MAC address first. If the host does not have the mapping in its DAI table, it creates an ARP request to resolve the mapping. All computers on the subnet will receive and process the ARP requests, and the host whose IP address matches the IP address in the request will send an ARP reply.

An ARP poisoning attack can target hosts, switches, and routers connected to the Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. For instance, a malicious host can reply to an ARP request with its own MAC address, thereby causing other hosts on the same subnet to store this information in their DAI tables or replace the existing ARP entry. Furthermore, a host can send gratuitous replies without having received any ARP requests. A malicious host can also send out ARP packets claiming to have an IP address that actually belongs to another host (e.g. the default router). After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

How DAI works

DAI allows only valid ARP requests and responses to be forwarded.

A device on which ARP Inspection is configured does the following:

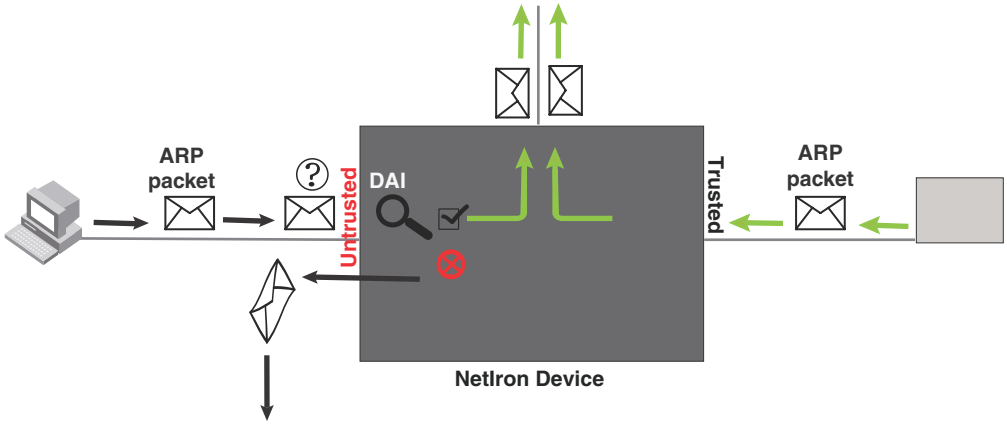
- Intercepts ARP packets received by the system CPU
- Inspects all ARP requests and responses received on untrusted ports
- Verifies that each of the intercepted packets has a valid IP-to-MAC address binding before updating the ARP table, or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

When you enable ARP Inspection on a VLAN, by default, all member ports are untrusted. You must manually configure trusted ports. In a typical network configuration, ports connected to host ports are untrusted. You configure ports connected to other switches or routers as trusted.

DAI inspects ARP packets received on untrusted ports, as shown in Figure 123. DAI carries out the inspection based on IP-to-MAC address bindings stored in a trusted binding database. For the PowerConnect device, the binding database is the ARP table, which supports DAI, DHCP snooping, and IP Source Guard. To inspect an ARP request packet, DAI checks the source IP and source MAC address against the ARP table. For an ARP reply packet, DAI checks the source IP, source MAC, destination IP, and destination MAC addresses. DAI forwards the valid packets and discards those with invalid IP-to-MAC address bindings.

When ARP packets reach a trusted port, DAI lets them through, as shown in Figure 123.

FIGURE 123 Dynamic ARP inspection at work



ARP entries

DAI uses the IP/MAC mappings in the ARP table to validate ARP packets received on untrusted ports. ARP entries in the ARP table derive from the following:

- **ARP Inspection** – statically configured VRF+VLAN +IP/MAC mapping.
- **ARP** – statically configured VRF+IP/MAC/port mapping.
- **DHCP-Snooping ARP** – information collected from snooping DHCP packets when DHCP snooping is enabled on VLANs.

Configuring DAI

NOTE
An index number is no longer needed to configure static ARP entries.

Follow the steps listed below to configure DAI.

1. Configure inspection of ARP entries for hosts on untrusted ports. Enable ARP Inspection on a VLAN to inspect ARP packets.
2. Configure the trust settings of the VLAN members. ARP packets received on *trusted* ports bypass the DAI validation process. ARP packets received on untrusted ports go through the DAI validation process.
3. Enable DHCP snooping to populate the DHCP snooping IP-to-MAC binding database. Refer to "DHCP binding database" on page 700 for more information.

The following shows the default settings of ARP Inspection.

Feature	Default
Dynamic ARP Inspection	Disabled
Trust setting for ports	Untrusted

Enabling dynamic ARP inspection on a VLAN

ARP and Dynamic inspection ARP entries need to be configured for hosts on untrusted ports. Otherwise, when Dynamic ARP Inspection checks ARP packets from these hosts against entries in the ARP table, it will not find any entries for them, and the device will not allow and learn ARP from an untrusted host.

Dynamic ARP Inspection is disabled by default. To enable Dynamic ARP Inspection on an existing VLAN or a range of VLANs, enter the following command.

```
NetIron(config)# ip arp-inspection vlan 18 to vlan 20
```

The command enables Dynamic ARP Inspection on VLAN 18 through VLAN 20. ARP packets from untrusted ports in VLAN 18 through VLAN 20 will undergo Dynamic ARP Inspection.

Syntax: [no] ip-arp inspection vlan <*vlan_id* > **to** <*vlan_id* >

The <*vlan_id*> variable specifies the ID of a configured VLAN or VLAN range. Valid VLAN ranges are 1-4090.

Configuring static ARP on a Vlan and port

In the PowerConnect configuration, the DHCP binding database is integrated with the ARP Inspection table. The ARP inspection table stores the DAI IP/MAC binding information, which is used to build the IP source guard ACL. The **static arp** command allows you to configure both the vlan id and port parameters on a layer 2 interface.

To configure a static arp entry for a vlan id, enter the following command.

```
NetIron(config)#arp 190.1.0.2 aabb.cc00.0100 vlan 10
```

Syntax: [no] arp <*ip*> <*mac*> **[vlan** <*vlan_id*>**]** [<*port*>]

The <*ip*> variable specifies the IP address for the static IP ARP entry.

The <*mac*> variable specifies the MAC address for the static IP ARP entry.

The <*vlan_id*> variable configures the static ARP entry for a vlan. The VLAN ID range is 1-4090.

The <*port*> variable configures the static ARP entry for a port.

If the vlan id is not configured when IP source guard is turned on, the IP address is assumed to be valid on all the vlans on the port.

If both the vlan id and the port are not configured when IP source guard is turned on, the IP address is assumed to be valid for all vlans.

Enabling trust on a port

The default trust setting for a port is untrusted. For ports that are connected to host ports, leave their trust settings as untrusted.

To enable trust on a port, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/4
NetIron(config-if-e10000-1/4)# arp-inspection-trust
```

The commands change the CLI to the interface configuration level of port 1/4 and set the trust setting of port 1/4 to trusted.

Syntax: [no] arp-inspection-trust

Creating ARP entries

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the PowerConnect router, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the device receives an ARP request from the device that has the entry's address.

To create a static ARP entry for a static MAC entry, enter a command such as the following.

```
NetIron(config)# arp 192.53.4.2 1245.7654.2348 vlan 10
```

The command adds a static ARP entry that maps IP address 192.53.4.2 to MAC address 1245.7654.2348. The entry is for a MAC address connected to VLAN 10 of the PowerConnect.

Syntax: [no] arp <*ip-addr*> <*mac-addr*> **{ ethernet** <*slot/port*> **| vlan** <*vlan_id*> **}**

The <*ip-addr*> parameter specifies the IP address of the device that has the MAC address of the entry.

The <*mac-addr*> parameter specifies the MAC address of the entry.

The **ethernet** <*slot/port*> command specifies the port number attached to the device that has the MAC address of the entry.

The **vlan** <*vlan_id*> variable specifies the ID of a configured VLAN or VLAN range. Valid VLAN ranges are 1-4090.

Creating a floating static ARP entry

You can create a static ARP entry without port assignments.

When a floating static ARP entry (Static ARP Inspection entry without port defined) is added to ARP Inspection table, the mapping is checked against the current static ARP table. If ARP entry with a matching IP but mismatch MAC is found, it will be deleted and a re-arp on the IP will be issued.

When an ARP entry is deleted from ARP Inspection table, the corresponding entry in the static ARP table will also be deleted.

To create a floating static ARP entry for a static MAC entry, enter a command such as the following.

```
NetIron(config)# arp 192.53.4.2 1245.7654.2348
```

The command adds a floating static ARP entry that maps IP address 192.53.4.2 to MAC address 1245.7654.2348.

Syntax: [no] arp <*ip-add*> <*mac-addr*> **[ethernet** <*portnum*> **| vlan** <*vlan_id*>**]**

The <*ip-addr*> parameter specifies the IP address of the device that has the MAC address of the entry.

The <*mac-addr*> parameter specifies the MAC address of the entry.

The **ethernet** <*portnum*> parameter specifies the port number attached to the device that has the MAC address of the entry, and is only valid for original static ARP entries.

The **vlan** <*vlan_id*> parameter specifies the ID of a configured VLAN.

Configuring a Virtual Routing Instance (VRF)

To configure a virtual routing instance (VRF), enter a command such as the following.

```
NetIron(config)# ip vrf vpn1
```

Syntax: [no] ip vrf <*vrf-name*>

The **ip vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Adding an ARP entry for a VRF

IP Addresses can be uniquely determined by VRF. The VLAN number is not needed because the VLAN information is obtained through the ARP protocol. To define an ARP inspection entry for a specific VRF, enter commands such as the following.

```
NetIron(config)# ip vrf vpn1
NetIron(config-ip-vrf-vpn1))#arp 192.53.4.2 1245.7654.2348 e 3/5
```

This command creates an ARP entry for vrf with IP address 192.53.4.2 and MAC address of 1245.7654.2348 on ethernet 3/5.

Syntax: [no] arp <*ip-addr*> <*mac-addr*> **[ethernet** <*slot/port*>**]**

The <*vrf-name*> parameter specifies the VRF you are configuring a static ARP entry for.

The <*ip-addr*> parameter specifies the IP address of the device that has the MAC address of the entry.

The <*mac-addr*> parameter specifies the MAC address of the entry.

The e**thernet** <*slot/port*> variable specifies the port number attached to the device that has the MAC address of the entry.

Displaying ARP inspection information

You can display ARP inspection information using the **show ip arp-inspection** and the **show ip static-arp** commands as shown in the following.

Displaying ARP inspection status and ports

To display the ARP inspection status for a VLAN and the trusted/untrusted ports in the VLAN, enter the following command.

```
NetIron# show ip arp-inspection
ARP inspected VLANs:
1000

ARP inspection trusted ports:
ethe 2/1
```

Syntax: show ip arp-inspection [vlan <*vlan_id*>**]**

The **vlan** <*vlan_id*> parameter specifies the ID of a configured VLAN.

Displaying ARP inspection statistics

You can use the **show ip arp-statistics** command to display ARP inspection counters for all ports on the router, as shown in the following.

```
NetIron# show ip arp-inspection-statistics
Module 1:
Port     Arp Packets Captured      Arp Packets Failed Inspection
1/1         0                        0
1/2         0                        0
1/3         0                        0
1/4         0                        0
1/5         0                        0
1/6         0                        0
1/7         0                        0
1/8         0                        0
1/9         0                        0
1/10        0                        0
1/11        0                        0
1/12        0                        0
1/13        0                        0
1/14        0                        0
1/15        0                        0
1/16        0                        0
1/17        0                        0
1/18        0                        0
1/19        0                        0
1/20        0                        0
Module 3:
Port     Arp Packets Captured      Arp Packets Failed Inspection
3/1         0                        0
3/2         0                        0
3/3         0                        0
3/4         690                      153
```

Specifying a port number with the **show ip arp-statistics** command displays the statistics for that port only, along with details of the last five ARP packets that failed inspection, as shown in the following.

```
NetIron# show ip arp-inspection-statistics ethernet 3/4
Arp packets captured: 695
Arp packets failed inspection: 158
Last 5 packets failed inspection:
Time                 Op  Target IP Target Mac     Source IP Source Mac     Vlan
2007-10-24  18:53:28 2   145.1.1.1 000c.dbe2.9353 145.1.1.2 0000.0900.0005 1
2007-10-24  18:53:29 2   145.1.1.1 000c.dbe2.9353 145.1.1.2 0000.0900.0005 1
2007-10-24  18:53:30 2   145.1.1.1 000c.dbe2.9353 145.1.1.2 0000.0900.0005 1
2007-10-24  18:53:32 2   145.1.1.1 000c.dbe2.9353 145.1.1.2 0000.0900.0005 1
2007-10-24  18:53:33 2   145.1.1.1 000c.dbe2.9353 145.1.1.2 0000.0900.0005 1
```

Syntax: show ip arp-inspection-statistics [slot <*slot-num*> **| ethernet** <*slot/port*> **]**

The **slot** option allows you to limit the display of ARP inspection statistics to the Ethernet interface module in the slot specified by the <*slot-num*> variable.

The **ethernet** option allows you to limit the display of ARP inspection statistics to the port specified by the <*slot/port*> variable. It also provides details of the last five ARP packets received by the specified port that failed inspection.

This display shows the following information.

TABLE 108 Show ip arp-inspection-statistics

This field...	Displays...
Port	The slot/port number.
Arp packets captured	The number of ARP packets captured for the specified port.
Arp packets failed inspection	The number of captured ARP packets that failed inspection for the specified port.
The following fields apply to the first five packets that failed inspection on the specified port.	
Time	The date and time that the packet was received on the port.
Op	The ARP operation mode.
Target IP	The destination IP address of the ARP rejected packet.
Target MAC	The destination MAC address of the ARP rejected packet.
Source IP	The source IP address of the ARP rejected packet.
Source MAC	The source MAC address of the ARP rejected packet.
VLAN	The VLAN number of the ARP rejected packet.

Clearing ARP inspection counters

You can use the clear arp-inspection-statistics command to clear the ARP inspection statistics counters for all ports on the router or for a specified module or port as shown in the following.

```
clear arp-inspection-statistics ethernet 3/1
```

Syntax: clear ip arp-inspection-statistics [slot <*slot-num*> **| ethernet** <*slot/port*> **]**

The **slot** option allows you to clear ARP inspection statistics for a single Ethernet interface module in a slot specified by the <*slot-num*> variable.

The **ethernet** option allows you to clear ARP inspection statistics for a single port specified by the <*slot/port*> variable.

Displaying the ARP table

To display the ARP Inspection table, enter the following command.

```
NetIron# show ip static-arp

Total no. of entries: 4
  Index   IP Address          MAC Address         Port      VLAN  ESI
  1       1.1.1.1             0001.0001.0001      1/1
  2       6.6.6.2             0002.0002.0002      1/2
  3       6.6.6.7             1111.1111.1111      2/1...
          Ports  : ethe 2/1 to 2/7 ethe 3/1 to 3/2
  4       7.7.7.7             0100.5e42.7f40      3/3
```

The command displays all ARP entries in the system.

Syntax: show ip static-arp

DHCP snooping

NOTE
DHCP snooping only supports IPV4 traffic.

Dynamic Host Configuration Protocol (DHCP) snooping enables the device to filter untrusted DHCP packets in a subnet. DHCP snooping can ward off MiM attacks, such as a malicious user posing as a DHCP server sending false DHCP server reply packets with the intention of misdirecting other users. DHCP snooping can also stop unauthorized DHCP servers and prevent errors due to user mis-configuration of DHCP servers.

NOTE
DHCP Snooping will not dynamically build the ARP Inspection table.

How DHCP snooping works

When enabled on a VLAN, DHCP snooping stands between untrusted ports (those connected to host ports) and trusted ports (those connected to DHCP servers). A VLAN with DHCP snooping enabled forwards DHCP request packets from clients and discards DHCP server reply packets on untrusted ports, and it forwards DHCP server reply packets on trusted ports to DHCP clients, as shown in the following figures.

FIGURE 124 DHCP snooping at work - on untrusted port

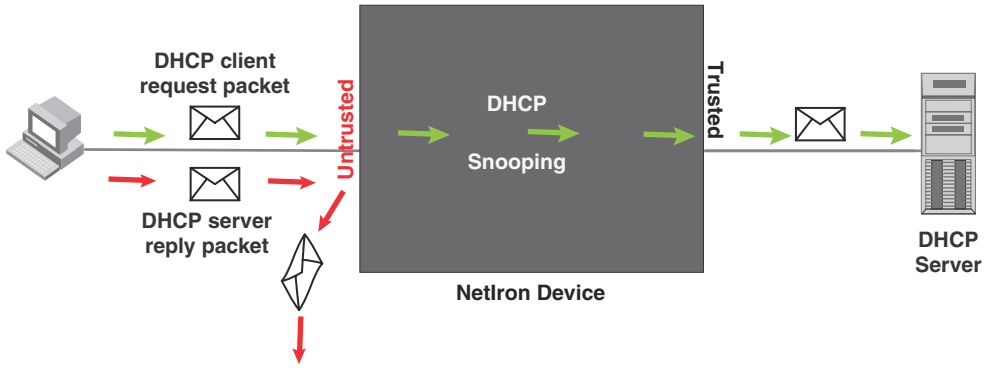
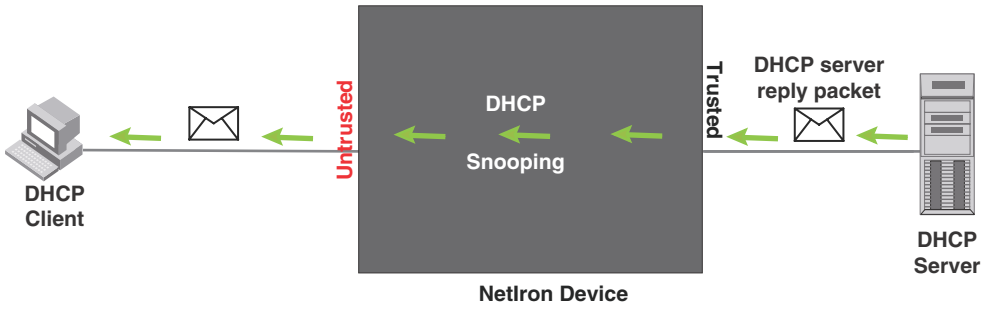


FIGURE 125 DHCP snooping at work - on trusted port



DHCP binding database

On trusted ports, DHCP server reply packets are forwarded to DHCP clients. The DHCP server reply packets collect client IP to MAC address binding information, which is saved in the DHCP binding database. This information includes MAC address, IP address, lease time, VLAN number, and port number.

In the PowerConnect configuration, the DHCP binding database is integrated with the enhanced ARP table, which is used by Dynamic ARP Inspection. For more information, refer to "ARP entries" on page 693.

The lease time will be refreshed when the client renews its IP address with the DHCP server; otherwise the router removes the entry when the lease time expires.

System reboot and the binding database

To allow DAI and DHCP snooping to work smoothly across a system reboot, the binding database is saved to a file in the system flash memory after the user issues the "reload" command. DHCP learned entries are written to the system flash memory before the router reboots. The flash file is written and read only if DHCP snooping is enabled.

Configuring DHCP snooping

Follow the steps listed below to configuring DHCP snooping.

1. Enable DHCP snooping on a VLAN.
2. For ports that are connected to a DHCP server, change their trust setting to trusted.

The following table shows the default settings of DHCP snooping:

Feature	Default
DHCP snooping	Disabled
Trust setting for ports	Untrusted

Enabling DHCP snooping on a VLAN

DHCP packets for a VLAN with DHCP snooping enabled are inspected.

DHCP snooping is disabled by default. This feature must be enabled on the client and the DHCP server VLANs. To enable DHCP snooping, enter the following global command for these VLANs.

```
NetIron(config)#ip dhcp-snooping vlan 2
```

The command enables DHCP snooping for a VLAN or a range of VLANS.

Syntax: [no] ip dhcp-snooping vlan <*vlan-number*> **[to** <*vlan_number*>**] [insert-relay-information]**

The <*vlan-number*> variable specifies the ID of a configured client or DHCP server VLAN.

If the [insert-relay-information] option is enabled, then DHCP option 82 is inserted in all the DHCP request packets. Refer to "DHCP binding database" on page 700 for more information.

Enabling trust on a port

The default trust setting for a port is untrusted. To enable trust on a port connected to a DHCP server, enter commands such as the following.

```
NetIron(config)#interface ethernet 1/1
NetIron(config-if-e10000-1/1)#dhcp-snooping-trust
```

Port 1/1 is connected to a DHCP server. The commands change the CLI to the interface configuration level of port 1/1 and set the trust setting of port 1/1 to trusted.

Syntax: [no] dhcp-snooping-trust

Clearing the DHCP binding database

You can clear the DHCP binding database using the **clear dhcp-binding** command. You can remove all entries in the database, or remove entries for a specific IP subnet, a VRF instance, or a VLAN id.

To remove all entries from the DHCP binding database, enter the following command.

```
NetIron#clear dhcp-binding
```

For example, to clear entries for a specific IP subnet, enter a command such as the following.

```
NetIron#clear dhcp 10.10.102.4
```

Syntax: clear dhcp [<*ip subnet*>] **[vlan** <*vlan_id*>] **[vrf** <*vrf_name*>]

The <*vlan_id*> variable specifies the ID of a configured VLAN.

The <*vrf_name*> variable specifies the VRF instance.

DHCP option 82 insertion

DHCP option 82 insertion can be used to assist DHCP servers to implement dynamic address policy. When DHCP option 82 is present in DHCP packets, DHCP servers gets additional information about the clients' identity.

The PowerConnect inserts DHCP option 82 when relaying DHCP request packets to DHCP servers. When DHCP server reply packets are forwarded back to DHCP clients, and sub-option 2 matches the local port MAC address, then DHCP option 82 is deleted. The vlan/port information is used to forward the DHCP reply. Refer to the following figures:

FIGURE 126 DHCP option 82 is added to the packet

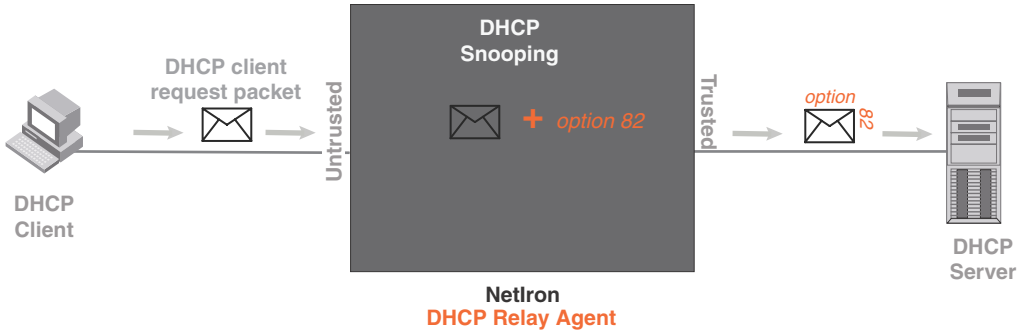
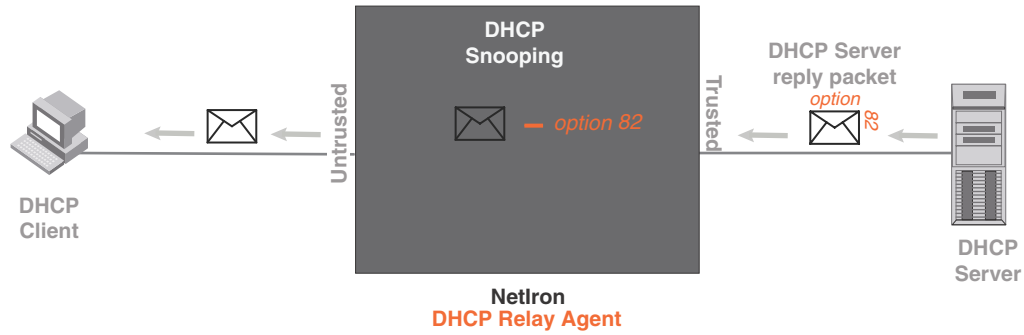


FIGURE 127 DHCP option 82 is removed from the packet



The option 82 insertion/deletion feature is available only when DHCP snooping is enabled for the client/server ports, and when the device is configured as a DHCP relay agent. By default, DHCP option 82 is off.

DHCP option 82 contains two sub-options; sub-option 1 (circuit ID) and sub-option 2 (remote ID).

Sub-option 1, relay agent circuit ID is in the following format.

VLAN id (2 bytes) / module id (1 byte) / port id (1 byte) (The module and port id will be 1 based).

The circuit ID identifies the location of the port, showing where the DHCP request comes from.

Typical address allocation is based on the gateway address of the relay agent.

Sub-option 2, Remote ID is in the following format.

MAC address (6 bytes)

Displaying DHCP snooping status and ports

To display the DHCP snooping status for a VLAN and the trusted and untrusted ports in the VLAN, enter the following command.

```
NetIron#show ip dhcp-snooping vlan 172
IP DHCP snooping VLAN 172: Enabled
  Trusted Ports : ethe 5/2 ethe 5/4
  Untrusted Ports : ethe 4/24 ethe 9/4 to 9/5 ethe 9/12 ethe 9/14
```

Syntax: show ip dhcp-snooping [vlan <*vlan-id*>**]**

Displaying DAI binding entries

To display all ARP inspection binding entries, including dhcp bindings specific to a VRF instance, enter the following command.

```
NetIron#(config)#show dai 201.1.1.0/24
Total no. of entries: 51
Idx Type IP Address      MAC Address       Port    Vlan Server IP    LTime
1   D    201.1.1.19      aabb.cc00.0012      10      200.1.1.2       3360
2   D    201.1.1.22      aabb.cc00.0007      10      200.1.1.2       3360
3   D    201.1.1.25      aabb.cc00.0030      10      200.1.1.2       3360
4   D    201.1.1.26      aabb.cc00.0004      10      200.1.1.2       3360
5   D    201.1.1.30      0030.488a.1c25      10      200.1.1.2       40200
6   D    201.1.1.32      aabb.cc00.0001      10      200.1.1.2       3360
7   D    201.1.1.34      aabb.cc00.0019      10      200.1.1.2       1560
8   D    201.1.1.39      aabb.cc00.000d      10      200.1.1.2       3360
9   D    201.1.1.44      aabb.cc00.0020      10      200.1.1.2       3360
10  D    201.1.1.46      aabb.cc00.0022          10       200.1.1.2       3360
```

Syntax: show dai [vrf <*vrf_name*>**] [vlan** <*vlan_id*>**] [**<*ip-subnet*>**]**

The <*vrf_name*> variable specifies the ARP entries that belong to a given VRF instance.

The <*vlan_id*> variable specifies the ID of a configured VLAN.

The <*ip subnet*> variable specifies the ARP entries that belong to specific IP-subnet address.

The following table describes the parameters of the **show dai** command:

TABLE 109 Display of show dai

This field...	Displays...
Index (Idx)	The row number of this entry in the IP route table.
Type	The ARP entry type, which can be any one of the following: Dynamic – The Layer 3 Switch learned the entry from an incoming packet. Static – The Layer 3 Switch loaded the entry from the static ARP table when the device for the entry was connected to the Layer 3 Switch. DHCP – The Layer 3 Switch learned the entry from the DHCP binding address table.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Port	The source port for the host vlan.
Vlan Server IP	The Vlan Server IP address of the server which assigns the IP/MAC mapping.
LTime	The lease time (aging timer for a DHCP entry).

Displaying DHCP snooping statistics counters

NOTE
The last five dropped packets are displayed through the CLI. Notifications and traps are not sent.

To display the DHCP snooping statistics counters, enter the following command.

```
NetIron#show ip dhcp-snooping-statistic slot 1
Module 1:
Port     DHCP Packets Captured      DHCP Packets dropped
1/1         0                        0
1/2         0                        0
1/3         0                        0
1/4         0                        0
1/5         0                        0
1/6         0                        0
1/7         0                        0
1/8         0                        0
1/9         0                        0
1/10        0                        0
1/11        0                        0
1/12        0                        0
1/13        0                        0
1/14        0                        0
1/15        0                        0
1/16        0                        0
1/17        0                        0
1/18        0                        0
1/19        9                        0
1/20        8                        6
```

The following table describes the output of the show ip dhcp snooping statisitc.

TABLE 110 Output from the show ip dhcp snooping statistic slot

This field...	Displays...
Module	Module number as positioned in the chassis.
Port	Port number in specified module.
DHCP Packets Captured	The number of DHCP packets captured on the port.
DHCP Packets Dropped	The number of DHCP packets dropped by DHCP snooping.

To display the DHCP snooping statistics counters for Ethernet ports, enter the following command.

```
NetIron#show ip dhcp-snooping-statistic eth 1/20
DHCP packets captured: 9
DHCP packets dropped by snooping: 7
Last 5 packets dropped by snooping:
Time                 DHCP type Source Mac/     Server IP/      Vlan
                               Source IP       Gateway IP
2008-05-03  00:29:43 OFFER     0030.4843.37ad  200.1.1.125     11
                               200.1.1.125     201.1.1.10
2008-05-03  00:29:59 OFFER     0030.4843.37ad  200.1.1.125     11
                               200.1.1.125     201.1.1.10
2008-05-03  00:31:18 OFFER     0030.4843.37ad  200.1.1.125     11
                               200.1.1.125     201.1.1.10
2008-05-03  00:31:22 OFFER     0030.4843.37ad  200.1.1.125     11
                               200.1.1.125     201.1.1.10
2008-05-03  00:31:30 OFFER     0030.4843.37ad  200.1.1.125     11
                               200.1.1.125     201.1.1.10
```

Syntax: show ip dhcp-snooping-statistic [slot <*slot*>**] | [ethernet** <*slot/port*>**]**

NOTE
If an Ethernet port is provided, the last five dropped packets are displayed

TABLE 111 Output from the show ip dhcp snooping statistic Ethernet port

This field...	Displays...
DHCP packets captured	The number of DHCP packets captured on port 20.
DHCP packets dropped by snooping	The number of DHCP packets dropped by DHCP snooping.
Last 5 packets dropped by snooping	The last 5 DHCP packets dropped per port.
Time	The time tracking system for collecting statistically information. Date and time are displayed.
DHCP Type	The DHCP Type displays the following: OFFER - When the server responds with a proposal of parameters. ACK- When the server assign an IP address. NAK- When the server rejects the request from the client.
Source MAC or Source IP	The Source MAC or Source IP address
Server IP or Gateway IP	The Serve IP or Gateway IP
Vlan	The VLAN number that DHCP Snooping was rejected on.

Clearing DHCP snooping counters

To clear the DHCP snooping statistic counters for a specific slot, enter the following command.

```
NetIron#clear dhcp-snooping-statistics slot 1
```

To clear the DHCP snooping statistic counters for a specific ethernet port, enter the following command.

```
NetIron#clear dhcp-snooping-statistics ethernet 1/20
```

Syntax: clear dhcp-snooping-statistics [slot <*slot*>] | [ethernet <*slot/port*>]

DHCP snooping configuration example

The following example configures VLAN 2 and VLAN 20, and changes the CLI to the global configuration level to enable DHCP snooping on the two VLANs. The commands are as follows.

```
NetIron(config)#vlan 2
NetIron(config-vlan-2)#untagged ethe 1/3 to 1/4
NetIron(config-vlan-2)#router-interface ve 2
NetIron(config-vlan-2)#exit
NetIron(config)# ip dhcp-snooping vlan 2

NetIron(config)#vlan 20
NetIron(config-vlan-20)#untagged ethe 1/1 to 1/2
NetIron(config-vlan-20)#router-interface ve 20
NetIron(config-vlan-20)#exit
NetIron(config)#ip dhcp-snooping vlan 20
```

On VLAN 2, client ports 1/3 and 1/4 are untrusted by default: all client ports are untrusted. Hence, only DHCP client request packets received on ports 1/3 and 1/4 are forwarded.

On VLAN 20, ports 1/1 and 1/2 are connected to a DHCP server. DHCP server ports are set to trusted.

```
NetIron(config)#interface ethernet 1/1
NetIron(config-if-e1000-1/1)#dhcp-snooping-trust
NetIron(config-if-e1000-1/1)#exit
NetIron(config)#interface ethernet 1/2
NetIron(config-if-e1000-1/2)#dhcp-snooping-trust
NetIron(config-if-e1000-1/2)#exit
```

Hence, DHCP sever reply packets received on ports 1/1 and 1/2 are forwarded, and client IP or MAC binding information is collected.

The example also sets the DHCP server address for the local relay agent.

```
NetIron(config)# interface ve 2
NetIron(config-vif-2)#ip address 20.20.20.1/24
NetIron(config-vif-2)#ip helper-address 30.30.30.4
NetIron(config-vif-2)#interface ve 20
NetIron(config-vif-20)#ip address 30.30.30.1/24
```

IP source guard

IP Source Guard permits traffic from only valid source IP addresses. IP source guard is used on client ports to prevent IP source address spoofing. Generally, IP source guard is used together with DHCP snooping and Dynamic ARP Inspection on untrusted ports. Refer to “DHCP snooping” on page 699 and “Dynamic ARP inspection” on page 691.

When IP Source Guard is first enabled, only DHCP packets are allowed and all other IP traffic is blocked. IP Source Guard uses IP or MAC bindings inside the ARP Inspection table to detect a valid IP address. When the system learns a valid IP address on the port, the client port then allows IP traffic to enter. If the source IP address of a packet does not match any of the IP addresses inside the ARP Inspection table, the packet is dropped. Only traffic with valid source IP addresses are permitted.

The system learns of a valid IP address from ARP. For information on how the ARP table is populated, Refer to“ARP entries” on page 693.

Enabling IP source guard

The **source-guard** command sets a port as an IP Source Guarded port. DHCP Snooping should be configured before you enable the IP Source Guard feature.

The default setting is disabled. To enable a port as an IP Source Guarded port, enter the following commands.

```
NetIron(config)# interface ethernet 2/2
NetIron(config-if-e10000-2/2)# source-guard
```

The commands change the CLI to the interface configuration level for port 2/2 and enable IP source guard on the port.

Syntax: [no] source-guard

NOTE
When IP Source Guard is enabled on a port it must have the same configuration as the primary port, otherwise it will not implemented as IP Source Guarded.

Enabling IP source inspection on a VLAN

IP Source Guard configuration is enabled on ports per vlan. When IP Source Guard is enabled on a vlan, by default all ports inside the vlan are set as “unguarded”. You can selectively turn on which ports inside the vlan to be set as “guarded”. Initially, when the vlan port is IP Source Guarded, only DHCP packets are allowed to get through. However, as IP or MAC binding is learned from DHCP snooping, or if it is manually configured, only packets with valid source IP address are allowed through.

There are two modes for IP Source Guard; strict mode and loose mode. You can configure either strict or loose mode during IP Source Guard vlan configuration. In a strict mode, the IP source address is bound to a particular port and vlan. Only packets with an IP address coming from a particular vlan port is considered valid. If the same source IP address is coming from a different port, then it is considered an attack and is dropped. The strict mode provides more security, but it does not allow for a layer 2 occurrence in a vlan. In a loose mode, the IP source address is bound to a vlan. Only packets with IP source addresses that come from ports within the vlan are considered valid.

To enable IP Source Inspection for a VLAN or a range of VLANs, enter the following command.

```
NetIron(config)# ip source-inspection vlan 2
```

Syntax: [no] ip source-inspection vlan <*vlan_number*> **[to** <*vlan_number*> **] [strict]**

The source IP addresses for VLAN IP packets are inspected for any port when IP Source Guard is enabled.

The <*vlan_number*> variable specifies the ID of a configured VLAN.

If the strict option is enabled, then valid IP source address is bound to a particular source port. This configuration can be learned from a DHCP reply, or manually configured.

NOTE
The strict mode requires DHCP relay-information insertion to be turned on.

Displaying IP source inspection status and ports

To display the IP Source Guard status for a VLAN, and the guarded or unguarded ports in the VLAN, enter the following command.

```
NetIron(config)#sh ip source-inspection vlan 10
IP Source Inspection configuration for VLAN 10:
Inspection mode: loose
un-guarded ports:
 ethe 1/4 ethe 1/18
guarded ports:
 ethe 1/20
```

The **show ip source-inspection vlan** command displays IP Source inspection configuration for VLAN 10 in loose mode.

Syntax: show ip source-inspection [vlan <*vlan_id*>**]**

The <*vlan_id*> variable specifies the ID of a configured vlan.

NOTE
This command is also available for debugging purposes on the Interface Module.

IP source guard CAM

The PowerConnect configuration uses a layer 4 ACL CAM to implement IP Source guard. When IP or MAC binding is learned or configured on an IP Source Guarded vlan-port, a layer 4 ACL CAM is programmed to allow valid source IP addresses.

When ACL is manually configured, a configuration conflict occurs with IP Source Guard, because it uses a layer 4 ACL CAM. The PowerConnect gives user ACL configuration a higher priority. When both IP Source Guard and user ACL is configured, the user ACL configuration takes precedence over IP Source Guard.

IP Source Guard uses layer 4 ACL CAM to check layer 2 switched traffic. When IP Source Guard is configured, the layer 3 port check flag is turned on. When IP Source Guard is configured, all traffic from the same physical port is subject to a layer 4 ACL check.

Configuring IP source guard CAM partition

IP Source Guard creates two CAM sub-partitions. The CAM sub-partitions include IP_SOURCE_GUARD_PERMIT and IP_SOURCE_GUARD_DENY. All CAM entries that are permitted, go to the IP_SOURCE_GUARD_PERMIT sub-partition. All CAM entries that are denied, go to the IP_SOURCE_GUARD_DENY sub-partition. The **system-max ip-source-guard-cam** command allows you to control the size of both IP_SOURCE_GUARD_PERMIT and IP_SOURCE_GUARD_DENY sub-partitions.

To specify a partition size of the IP Source Guard CAM, enter the following command.

```
NetIron(config)#system-max ip-source-guard-cam 1008
```

Syntax: [no] system-max [ip-source-guard-cam <*decimal*>**]**

By default, **no** system-max is configured.

The <*decimal*> variable specifies the range that is supported for configuring IP Source Guard CAM sub-partitions. The decimal range is from 0 to 131072. The default is 0.

Displaying IP source guard CAM partition

To display all IP Source Guard CAM partition on a layer 4 interface, enter the following command.

```
NetIron(config)# show cam l4 2/1
LP Index   Src  IP   SPort  Pro  Age IFL/ Out IF Group PRAM
   (Hex)(Dest IP         DPort)        VLAN Action       (Hex)
2  52000 0.0.0.0         0     17  Dis 0    CPU    31    00084
        (127.0.0.0       3784 )
```

Syntax: show cam <*interface*>

The following table describes the output of the **show cam** command.

TABLE 112 Display of show cam command

This field...	Displays...
Index (Hex)	The row number of this entry in the IP route table
Src IP/ Dest IP	The Source IP address or Destination IP address
SPort or DPort	The Source Port or Destination Port
Pro	The type of protocol (TCP, UDP) used.
Age	The Age is disabled.
IFL or VLAN	The VLAN that the port belongs to.
Action	The type of action: Pass or Drop.

Configuring forwarding parameters

The following configurable parameters control the forwarding behavior of the PowerConnect:

- Time-To-Live (TTL) threshold
- Forwarding of directed broadcasts
- Forwarding of source-routed packets
- Ones-based and zero-based broadcasts

All these parameters are global and thus affect all IP interfaces configured on the PowerConnect.

To configure these parameters, use the procedures in the following sections.

Changing the TTL threshold

The TTL threshold prevents routing loops by specifying the maximum number of router hops an IP packet originated by the PowerConnect can travel through. Each device capable of forwarding IP that receives the packet decreases the packet's TTL by one. If a device receives a packet with a TTL of 1 and reduces the TTL to zero, the device drops the packet.

The default TTL is 64. You can change the TTL to a value from 1– 255.

To modify the TTL threshold to 25, enter the following commands.

```
NetIron(config)# ip ttl 25
```

Syntax: [no] ip ttl <*1-255*>

Enabling forwarding of directed broadcasts

A directed broadcast is an IP broadcast to all devices within a single directly-attached network or subnet. A net-directed broadcast goes to all devices on a given network. A subnet-directed broadcast goes to all devices within a given subnet.

NOTE
A less common type, the all-subnets broadcast, goes to all directly-attached subnets. Forwarding for this broadcast type also is supported, but most networks use IP multicasting instead of all-subnet broadcasting.

Forwarding for all types of IP directed broadcasts is disabled by default. You can enable forwarding for all types if needed. You cannot enable forwarding for specific broadcast types.

To enable forwarding of IP directed broadcasts, enter the following command.

```
NetIron(config)# ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

The software makes the forwarding decision based on the router's knowledge of the destination network prefix. Routers cannot determine that a message is unicast or directed broadcast apart from the destination network prefix. The decision to forward or not forward the message is by definition only possible in the last hop router.

To disable the directed broadcasts, enter the following command in the CONFIG mode.

```
NetIron(config)# no ip directed-broadcast
```

To enable directed broadcasts on an individual interface instead of globally for all interfaces, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

Disabling forwarding of IP source-routed packets

A source-routed packet specifies the exact router path for the packet. The packet specifies the path by listing the IP addresses of the router interfaces through which the packet must pass on its way to the destination. The PowerConnect supports both types of IP source routing:

- **Strict source routing** – requires the packet to pass through only the listed routers. If the PowerConnect receives a strict source-routed packet but cannot reach the next hop interface specified by the packet, the PowerConnect discards the packet and sends an ICMP Source-Route-Failure message to the sender.

  NOTE
  The PowerConnect allows you to disable sending of the Source-Route-Failure messages. Refer to "Disabling ICMP messages" on page 712.

- **Loose source routing** – requires that the packet pass through all of the listed routers but also allows the packet to travel through other routers, which are not listed in the packet.

The PowerConnect forwards both types of source-routed packets by default. You cannot enable or disable strict or loose source routing separately.

To disable forwarding of IP source-routed packets, enter the following command.

```
NetIron(config)# no ip source-route
```

Syntax: [no] ip source-route

To re-enable forwarding of source-routed packets, enter the following command.

```
NetIron(config)# ip source-route
```

Enabling support for zero-based IP subnet broadcasts

By default, the PowerConnect treats IP packets with all ones in the host portion of the address as IP broadcast packets. For example, the PowerConnect treats IP packets with 209.157.22.255/24 as the destination IP address as IP broadcast packets and forwards the packets to all IP hosts within the 209.157.22.x subnet (except the host that sent the broadcast packet to the PowerConnect).

Most IP hosts are configured to receive IP subnet broadcast packets with all ones in the host portion of the address. However, some older IP hosts instead expect IP subnet broadcast packets that have all zeros instead of all ones in the host portion of the address. To accommodate this type of host, you can enable the PowerConnect to treat IP packets with all zeros in the host portion of the destination IP address as broadcast packets.

NOTE
When you enable the PowerConnect for zero-based subnet broadcasts, the PowerConnect still treats IP packets with all ones the host portion as IP subnet broadcasts too. Thus, the PowerConnect can be configured to support all ones only (the default) or all ones and all zeroes.

NOTE
This feature applies only to IP subnet broadcasts, not to local network broadcasts. The local network broadcast address is still expected to be all ones.

To enable the PowerConnect for zero-based IP subnet broadcasts in addition to ones-based IP subnet broadcasts, enter the following command.

```
NetIron(config)# ip broadcast-zero
```

Syntax: [no] ip broadcast-zero

Configuring the maximum ICMP error message rate

NOTE
The maximum ICMP error message rate configuration only supports IPv4 traffic.

The PowerConnect configuration allows 200 ICMP error messages per second per IP interface. You can now configure the maximum ICMP error message rate on all Interface Modules. The maximum configured value is increased to 5000 error messages per second. The maximum ICMP error message rate configuration uses an ICMP error metering mechanism. The process for the ICMP error metering mechanism is as follows:

- There is a meter counter for each interface. There is one total meter counter per Interface Module.
- The interface counter and the total counter will increment every time an icmp error message is sent out.
- The timer will rest all counters to 0 every second.
- Before an error message is sent out, it check the interface meter counter against the user configured icmp error limit (5000 max). The total counter will check against 10000. The error message is dropped if one any counter is larger the checked value.

The total error rate for all IP interfaces on an Interface Module is 10,000 errors per second. The ICMP error metering mechanism is per IP interface; this includes VRF IP interfaces.

Since the ICMP error metering code implementation is similar between the Management Module and Interface Module code, this change will also affect the Management Module ICMP error rate.

To configure the maximum ICMP error rate, enter the following command.

```
NetIron(config)# ip icmp max-err-msg-rate 600
```

Syntax: [no] ip icmp max-err-msg-rate <*error per second*>

The <*error per second*> variable specifies the maximum error rate in errors per second. The maximum configured value has a range from 0 (minimum) to 5000 (maximum) error message per second. The default value is 400.

Disabling ICMP messages

The PowerConnect is enabled to reply to ICMP echo messages and send ICMP Destination Unreachable messages by default.

You can selectively disable the following types of Internet Control Message Protocol (ICMP) messages:

- Echo messages (ping messages) – The PowerConnect replies to IP pings from other IP devices.
- Destination Unreachable messages – If the PowerConnect receives an IP packet that it cannot deliver to its destination, the PowerConnect discards the packet and sends a message back to the device that sent the packet. The message informfs the device that the destination cannot be reached by the PowerConnect.

Disabling replies to broadcast ping requests

By default, the PowerConnect is enabled to respond to broadcast ICMP echo packets, which are ping requests.

To disable response to broadcast ICMP echo packets (ping requests), enter the following command.

```
NetIron(config)# no ip icmp echo broadcast-request
```

Syntax: [no] ip icmp echo broadcast-request

If you need to re-enable response to ping requests, enter the following command.

```
NetIron(config)# ip icmp echo broadcast-request
```

Disabling ICMP destination unreachable messages

By default, when the PowerConnect receives an IP packet that the device cannot deliver, the device sends an ICMP Unreachable message back to the host that sent the packet. You can selectively disable a PowerConnect's response to the following types of ICMP Unreachable messages:

- Administration – The packet was dropped by the device due to a filter or ACL configured on the device.
- Fragmentation-needed – The packet has the Do not Fragment bit set in the IP Flag field, but the PowerConnect cannot forward the packet without fragmenting it.
- Host – The destination network or subnet of the packet is directly connected to the PowerConnect, but the host specified in the destination IP address of the packet is not on the network.
- Network – The PowerConnect cannot reach the network specified in the destination IP address of the packet.

- **Port** – The destination host does not have the destination TCP or UDP port specified in the packet. In this case, the host sends the ICMP Port Unreachable message to the PowerConnect, which in turn sends the message to the host that sent the packet.
- **Protocol** – The TCP or UDP protocol on the destination host is not running. This message is different from the Port Unreachable message, which indicates that the protocol is running on the host but the requested protocol port is unavailable.
- **Source-route-failure** – The device received a source-routed packet but cannot locate the next-hop IP address indicated in the packet's Source-Route option.

You can disable the PowerConnect from sending these types of ICMP messages on an individual basis.

NOTE
Disabling an ICMP Unreachable message type does not change the PowerConnect's ability to forward packets. Disabling ICMP Unreachable messages prevents the device from generating or forwarding the Unreachable messages.

To disable all ICMP Unreachable messages, enter the following command.

```
NetIron(config)# no ip icmp unreachable
```

Syntax: **[no] ip icmp unreachable [network | host | protocol | administration | fragmentation-needed | port | source-route-fail]**

- If you enter the command without specifying a message type (as in the example above), all types of ICMP Unreachable messages listed above are disabled. If you want to disable only specific types of ICMP Unreachable messages, you can specify the message type. To disable more than one type of ICMP message, enter the **no ip icmp unreachable** command for each messages type.
- The **network** parameter disables ICMP Network Unreachable messages.
- The **host** parameter disables ICMP Host Unreachable messages.
- The **protocol** parameter disables ICMP Protocol Unreachable messages.
- The **administration** parameter disables ICMP Unreachable (caused by Administration action) messages.
- The **fragmentation-needed** parameter disables ICMP Fragmentation-Needed But Do not-Fragment Bit Set messages.
- The **port** parameter disables ICMP Port Unreachable messages.
- The **source-route-fail** parameter disables ICMP Unreachable (caused by Source-Route-Failure) messages.

To disable ICMP Host Unreachable messages and ICMP Network Unreachable messages but leave the other types of ICMP Unreachable messages enabled, enter the following commands instead of the command shown above.

```
NetIron(config)# no ip icmp unreachable host
NetIron(config)# no ip icmp unreachable network
```

If you have disabled all ICMP Unreachable message types but want to re-enable certain types, you can do so by entering commands such as the following.

```
NetIron(config)# ip icmp unreachable host
NetIron(config)# ip icmp unreachable network
```

These commands re-enable ICMP Unreachable Host messages and ICMP Network Unreachable messages.

Disabling ICMP redirect messages

You can disable or re-enable ICMP redirect messages. By default, the PowerConnect sends an ICMP redirect message to the source of a misdirected packet in addition to forwarding the packet to the appropriate router. You can disable ICMP redirect messages on a global basis or on an individual port basis.

NOTE
An unusually high receipt of multiple Internet Control Message Protocol (ICMP) Redirect packets that are used to change routing table entries in a short period of time may cause high CPU utilization. However this can be avoided by configuring the maximum ICMP error message rate using **ip icmp max-err-msg-rate** command, 0 (minimum) to 5000 (maximum) error message per second. The default value is 400.The total error rate for all IP interfaces (SYSTEM) is 10,000 errors per second.

NOTE
The device forwards misdirected traffic to the appropriate router, even if you disable the redirect messages.

To disable ICMP redirect messages globally, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# no ip icmp redirects
```

Syntax: [no] ip icmp redirects

To disable ICMP redirect messages on a specific interface, enter the following command at the configuration level for the interface.

```
NetIron(config)# int e 3/11
NetIron(config-if-e100-3/11)# no ip redirect
```

Syntax: [no] ip redirect

Configuring static routes

The IP route table can receive routes from the following sources:

- **Directly-connected networks** – When you add an IP interface, the PowerConnect automatically creates a route for the network the interface is in.
- **RIP** – If RIP is enabled, the PowerConnect can learn about routes from the advertisements other RIP routers send to the PowerConnect. If the route has a lower administrative distance than any other routes from different sources to the same destination, the PowerConnect places the route in the IP route table.
- **OSPF** – Refer to RIP, but substitute “OSPF” for “RIP”.
- **BGP4** – Refer to RIP, but substitute “BGP4” for “RIP”.
- **Default network route** – A statically configured default route that the PowerConnect uses if other default routes to the destination are not available. Refer to “Configuring a default network route” on page 730.
- **Statically configured route** – You can add routes directly to the route table. When you add a route to the IP route table, you are creating a static IP route. This section describes how to add static routes to the IP route table.

Static route types

You can configure the following types of static IP routes:

- **Standard** – the static route consists of the destination network address and network mask, and the IP address of the next-hop gateway. You can configure multiple standard static routes with the same metric for load sharing or with different metrics to provide a primary route and backup routes.
- **Interface-based** – the static route consists of the destination network address and network mask, and the PowerConnect interface through which you want the PowerConnect to send traffic for the route. Typically, this type of static route is for directly attached destination networks.
- **Null** – the static route consists of the destination network address and network mask, and the "null0" parameter. Typically, the null route is configured as a backup route for discarding traffic if the primary route is unavailable.

Static IP route parameters

When you configure a static IP route, you must specify the following parameters:

- The IP address and network mask for the route's destination network.
- The route's path, which can be one of the following:
  - The IP address of a next-hop gateway
  - An Ethernet port
  - A virtual interface (a routing interface used by VLANs for routing Layer 3 protocol traffic among one another)
  - A "null" interface. The PowerConnect drops traffic forwarded to the null interface.

The following parameters are optional:

- **The route's metric** – The value the PowerConnect uses when comparing this route to other routes in the IP route table to the same destination. The metric applies only to routes that the PowerConnect has already placed in the IP route table. The default metric for static IP routes is 1.
- **The route's administrative distance** – The value that the PowerConnect uses to compare this route with routes from other route sources to the same destination before placing a route in the IP route table. This parameter does not apply to routes that are already in the IP route table. The default administrative distance for static IP routes is 1.

The default metric and administrative distance values ensure that the PowerConnect always prefers static IP routes over routes from other sources to the same destination.

Multiple static routes to the same destination provide load sharing and redundancy

You can add multiple static routes for the same destination network to provide one or more of the following benefits:

- **IP load balancing** – When you add multiple IP static routes for the same destination to different next-hop gateways, and the routes each have the same metric and administrative distance, the PowerConnect can load balance traffic to the routes' destination. For information about IP load balancing, refer to "Configuring IP load sharing" on page 731.

- **Path redundancy** – When you add multiple static IP routes for the same destination, but give the routes different metrics or administrative distances, the PowerConnect uses the route with the lowest administrative distance by default, but uses another route to the same destination of the first route becomes unavailable.

Refer to the following sections for examples and configuration information:

- "Configuring load balancing and redundancy using multiple static routes to the same destination" on page 721
- "Configuring standard static IP routes and interface or null static routes to the same destination" on page 722

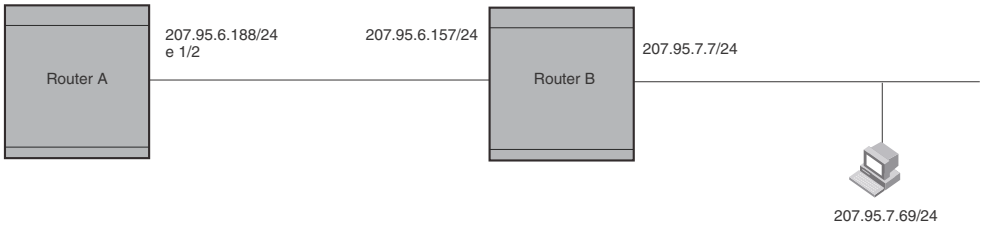
Static route states follow port states

IP static routes remain in the IP route table only so long as the port or virtual interface used by the route is available. If the port or virtual routing interface becomes unavailable, the software removes the static route from the IP route table. If the port or virtual routing interface becomes available again later, the software adds the route back to the route table.

This feature allows the PowerConnect to adjust to changes in network topology. The PowerConnect does not continue trying to use routes on unavailable paths but instead uses routes only when their paths are available.

Figure 128 shows a network containing a static route. The static route is configured on Router A, as shown in the CLI following the figure.

FIGURE 128 Example of a static route



The following command configures a static route to 207.95.7.0, using 207.95.6.157 as the next-hop gateway.

```
NetIron(config)# ip route 207.95.7.0/24 207.95.6.157
```

When you configure a static IP route, you specify the destination address for the route and the next-hop gateway or PowerConnect interface through which the PowerConnect can reach the route. The PowerConnect adds the route to the IP route table. In this case, Router A knows that 207.95.6.157 is reachable through port 1/2, and also assumes that local interfaces within that subnet are on the same port. Router A deduces that IP interface 207.95.7.188 is also on port 1/2.

The software automatically removes a static IP route from the IP route table if the port used by that route becomes unavailable. When the port becomes available again, the software automatically re-adds the route to the IP route table.

Configuring a static IP route

To configure an IP static route with a destination address of 192.0.0.0 255.0.0.0 and a next-hop router IP address of 195.1.1.1, enter the following.

```
NetIron(config)# ip route 192.0.0.0 255.0.0.0 195.1.1.1
```

To configure a default route, enter the following.

```
NetIron(config)# ip route 0.0.0.0  0.0.0.0
```

To configure a static IP route with an Ethernet port instead of a next-hop address, enter a command such as the following.

```
NetIron(config)# ip route 192.128.2.69 255.255.255.0 ethernet 4/1
```

The command configures a static IP route for destination network 192.128.2.69/24. Since an Ethernet port is specified instead of a gateway IP address as the next hop, the PowerConnect always forwards traffic for the 192.128.2.69/24 network to port 4/1.

To configure an IP static route that uses virtual interface 3 as its next hop, enter a command such as the following.

```
NetIron(config)# ip route 192.128.2.71 255.255.255.0 ve 3
```

Syntax: **ip route** <*dest-ip-addr*> <*dest-mask*> | <*dest-ip-addr*>/<*mask-bits*> <*next-hop-ip-addr*> | **ethernet** <*slot/port*> | **pos** <*slot/port*> | **ve** <*num*> *[*<*metric*>**]** **[tag** <*num*>**]** **[distance** <*num*>**]** **[name** <*string*>**]**

The <*dest-ip-addr*> is the route's destination. The <*dest-mask*> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering / followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24.

The <*next-hop-ip-addr*> is the IP address of the next-hop router (gateway) for the route.

For a default route, enter 0.0.0.0 0.0.0.0 xxx.xxx.xxx.xxx (use 0 for the <*mask-bits*> if you specify the address in CIDR format).

If you do not want to specify a next-hop IP address, you can instead specify a port or interface number on the PowerConnect. The <*num*> parameter is a virtual interface number. The <*slot/port*> is the port's number of the PowerConnect. If you specify an Ethernet port, the PowerConnect forwards packets destined for the static route's destination network to the specified interface. Conceptually, this feature makes the destination network like a directly connected network, associated with a PowerConnect interface.

NOTE
The port or virtual interface you use for the static route's next hop must have at least one IP address configured on it. The address does not need to be in the same subnet as the destination network.

The <*metric*> parameter specifies the cost of the route and can be a number from 1 – 16. The default is 1.

NOTE
If you specify 16, RIP considers the metric to be infinite and thus also considers the route to be unreachable.

The **tag** <*num*> parameter specifies the tag value of the route. Possible values: 0 - 4294967295. Default: 0.

The **distance** <*num*> parameter specifies the administrative distance of the route. When comparing otherwise equal routes to a destination, the PowerConnect prefers lower administrative distances over higher ones, so make sure you use a low value for your default route. Possible values: 1 - 255. Default: 1.

NOTE
The PowerConnect will replace the static route if it receives a route with a lower administrative distance.

The **name** <*string*> parameter specifies the name assigned to a route. The static route name is descriptive and an optional feature. It does not affect the selection of static routes.

Configuring a static IP route between VRFs

You can configure a static route next hop to be in a different VRF. This can be done for the following:

- From the default VRF to a non-default VRF
- From a non-default VRF to a non-default VRF
- From a non-default VRF to the default VRF
- From one VRF to an IP interface in a different VRF.

NOTE
RPF is not supported with the Static Route between VRFs feature.

NOTE
For information on disabling gratuitous ARP requests on a VRF IP interface, refer to "Disabling gratuitous ARP requests for local proxy ARP" on page 690.

Configuring a static route from the default VRF to a non-default VRF

To configure an IP static route with a destination address of 192.0.0.0/24 and a next-hop router with an IP address of 195.1.1.1 in the non-default VRF named "blue", enter the following at the general configuration prompt.

```
NetIron(config)# ip route 192.128.2.69/24 next-hop-vrf blue 195.1.1.1
```

Syntax: **[no] ip route** <*dest-ip-addr*> <*dest-mask*> | <*dest-ip-addr*>/<*mask-bits*>
next-hop-vrf <*next-hop-vrf-name*> <*next-hop-ip-addr*>

The <*dest-ip-addr*> is the route's destination. The <*dest-mask*> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering / followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24.

The <*next-hop-vrf-name*> is the name of the VRF that contains the next-hop router (gateway) for the route.

The <*next-hop-ip-addr*> is the IP address of the next-hop router (gateway) for the route.

NOTE
The **next-hop-vrf** needs to be a valid VRF to be used in this command.

Configuring a static route from a non-default VRF to a non-default VRF

To configure an IP static route within the VRF named "red" with a destination address of 192.0.0.0/24 and a next-hop router with an IP address of 195.1.1.1 in the non-default VRF named "blue", enter the following from within the VRF "red" configuration context.

```
NetIron(config)# ip vrf red
NetIron(config-vrf-red)# ip route 192.128.2.69/24 next-hop-vrf blue 195.1.1.1
```

Syntax: **[no] ip route** <*dest-ip-addr*> <*dest-mask*> | <*dest-ip-addr*>/<*mask-bits*>
next-hop-vrf <*next-hop-vrf-name*> <*next-hop-ip-addr*>

The <*dest-ip-addr*> is the route's destination. The <*dest-mask*> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering / followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24.

The <*next-hop-vrf-name*> is the name of the VRF that contains the next-hop router (gateway) for the route.

The <*next-hop-ip-addr*> is the IP address of the next-hop router (gateway) for the route.

NOTE
The **next-hop-vrf** needs to be a valid VRF to be used in this command.

Configuring a static route from a non-default VRF to the default VRF

To configure an IP static route within the VRF named "red" with a destination address of 192.0.0.0/24 and a next-hop router in the default VRF and an IP address of 195.1.1.1, enter the following from within the VRF "red" configuration context.

```
NetIron(config)# ip vrf red
NetIron(config-vrf-red)# ip route 192.128.2.69/24 next-hop-vrf default-vrf
195.1.1.1
```

Syntax: **[no] ip route** <*dest-ip-addr*> <*dest-mask*> | <*dest-ip-addr*>/<*mask-bits*>
next-hop-vrf <*next-hop-vrf-name*> <*next-hop-ip-addr*>

The <*dest-ip-addr*> is the route's destination. The <*dest-mask*> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering / followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24.

The **default-vrf** option specifies that the next-hop router (gateway) for the route is in the default VRF.

The <*next-hop-ip-addr*> is the IP address of the next-hop router (gateway) for the route.

NOTE
The **next-hop-vrf** needs to be a valid VRF to be used in this command.

Configuring an IP static interface route across VRFs

You can configure an IP Static interface route from one VRF to an IP interface in a different VRF. This allows you to connect from one VRF to a host that is directly connected to a port in a different VRF. You can do this by configuring a static route to point to the interface that is directly connected to the device with the IP address you want to reach. The following example defines two VRFs as follows:

VRF A :

Route Distinguiser = 1:1

Interface: ethernet port 1/1

IP address: 10.0.0.1/24

VRF B :

Route Distinguiser = 2:2

Interface: ethernet port 1/2

IP address: 20.0.0.1/24

```
NetIron(config)# ip vrf A
NetIron(config-vrf-A)# rd 1:1
NetIron(config-vrf-A)# exit
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# ip vrf forwarding A
NetIron(config-if-e10000-1/1)# ip address 10.0.0.1/24
NetIron(config-if-e10000-1/1)# exit
NetIron(config)# ip vrf B
NetIron(config-vrf-B)# rd 2:2
NetIron(config-vrf-B)# exit
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e10000-1/2)# ip vrf forwarding B
NetIron(config-if-e10000-1/2)# ip address 20.0.0.1/24
```

The following example configures an IP Static interface route from VRF A to a network with IP address 20.0.0.0/24, which is directly connected to ethernet port 1/2 in VRF B:

```
NetIron(config)# ip vrf a
NetIron(config-vrf-a)# ip route 20.0.0.0/24 ethernet 1/2
```

Syntax: **[no] ip route** <*dest-ip-addr*>/<*mask-bits*> **[ethernet** <*slot/port*> **| ve** <*num*> **]**

The <*dest-ip-addr*> is the route's destination. The <*dest-mask*> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24. To configure a default route, enter 0.0.0.0 for <*dest-ip-addr*> and 0.0.0.0 for <*dest-mask*> (or 0 for the <*mask-bits*> if you specify the address in CIDR format). Specify the IP address of the default gateway using the <*next-hop-ipaddr*> parameter.

The <*slot/port*> or <*num*> is an interface in a different VRF that is directly connected to the device that you want to reach.

Configuring a "null" route

You can configure the PowerConnect to drop IP packets to a specific network or host address by configuring a "null" (sometimes called "null0") static route for the address. When the PowerConnect receives a packet destined for the address, the PowerConnect drops the packet instead of forwarding it.

To configure a null static route to drop packets destined for network 209.157.22.x, enter the following commands.

```
NetIron(config)# ip route 209.157.22.0 255.255.255.0 null0
NetIron(config)# write memory
```

Syntax: **[no] ip route** <*ip-addr*> <*ip-mask*> **|** <*dest-ip-addr*>/<*mask-bits*> **null0 [**<*metric*>*] [tag* <*num*>**] [distance** <*num*>**]**

To display the maximum value for your device, enter the **show default values** command. The maximum number of static IP routes the system can hold is listed in the ip-static-route row in the System Parameters section of the display. To change the maximum value, use the **system-max ip-static-route** <*num*> command at the global CONFIG level.

The <*ip-addr*> parameter specifies the network or host address. The PowerConnect will drop packets that contain this address in the destination field instead of forwarding them.

The <*ip-mask*> parameter specifies the network mask. Ones are significant bits and zeros allow any value. For example, the mask 255.255.255.0 matches on all hosts within the Class C subnet address specified by <*ip-addr*>. Alternatively, you can specify the number of bits in the network mask. For example, you can enter 209.157.22.0/24 instead of 209.157.22.0 255.255.255.0.

The **null0** parameter indicates that this is a null route. You must specify this parameter to make this a null route.

The <*metric*> parameter adds a cost to the route. You can specify from 1 – 16. The default is 1.

The **tag** <*num*> parameter specifies the tag value of the route. Possible values: 0 - 4294967295. Default: 0.

The **distance** <*num*> parameter configures the administrative distance for the route. You can specify a value from 1 – 255. The default is 1. The value 255 makes the route unusable.

NOTE
The last three parameters are optional and do not affect the null route, unless you configure the administrative distance to be 255. In this case, the route is not used and the traffic might be forwarded instead of dropped.

Dropping traffic sent to the null0 interface In hardware

Traffic sent to the null0 interface is done in hardware; that is, by programming the CAM to discard traffic sent to the null0 interface. This improves forwarding efficiency and reduces the burden on the PowerConnect's CPU.

Hardware dropping for IP traffic sent to the null0 interface is supported.

You can optionally configure the PowerConnect to drop traffic sent to the default IP route address in hardware. To do this, enter the following commands.

```
NetIron(config)# ip route 0.0.0.0 0.0.0.0 null0
NetIron(config)# ip hw-drop-on-def-route
```

Syntax: [no] ip hw-drop-on-def-route

CAM default route aggregation

Configuring the PowerConnect to drop traffic sent to the default IP route address in hardware causes the device to program 32-bit host CAM entries for each destination address using the default route, which could consume the CAM space. To prevent this from happening, you can enable the CAM Default Route Aggregation feature. To do this, enter the following command.

```
NetIron(config)# ip dr-aggregate
```

Syntax: [no] ip dr-aggregate

Configuring load balancing and redundancy using multiple static routes to the same destination

You can configure multiple static IP routes to the same destination, for the following benefits:

- **IP load sharing** – If you configure more than one static route to the same destination, and the routes have different next-hop gateways but have the same metrics, the PowerConnect load balances among the routes using basic round-robin. For example, if you configure two static routes with the same metrics but to different gateways, the PowerConnect alternates between the two routes. For information about IP load balancing, refer to "Configuring IP load sharing" on page 731.

- **Backup Routes** – If you configure multiple static IP routes to the same destination, but give the routes different next-hop gateways and different metrics, the PowerConnect will always use the route with the lowest metric. If this route becomes unavailable, the PowerConnect will fail over to the static route with the next-lowest metric, and so on.

NOTE
You also can bias the PowerConnect to select one of the routes by configuring them with different administrative distances. However, make sure you do not give a static route a higher administrative distance than other types of routes, unless you want those other types to be preferred over the static route.

The steps for configuring the static routes are the same as described in the previous section. The following sections provide examples.

To configure multiple static IP routes, enter commands such as the following.

```
NetIron(config)# ip route 192.128.2.69 255.255.255.0 209.157.22.1
NetIron(config)# ip route 192.128.2.69 255.255.255.0 192.111.10.1
```

The commands in the example above configure two static IP routes. The routes go to different next-hop gateways but have the same metrics. These commands use the default metric value (1), so the metric is not specified. These static routes are used for load sharing among the next-hop gateways.

The following commands configure static IP routes to the same destination, but with different metrics. The route with the lowest metric is used by default. The other routes are backups in case the first route becomes unavailable. The PowerConnect uses the route with the lowest metric if the route is available.

```
NetIron(config)# ip route 192.128.2.69 255.255.255.0 209.157.22.1
NetIron(config)# ip route 192.128.2.69 255.255.255.0 192.111.10.1 2
NetIron(config)# ip route 192.128.2.69 255.255.255.0 201.1.1.1 3
```

In this example, each static route has a different metric. The metric is not specified for the first route, so the default (1) is used. A metric is specified for the second and third static IP routes. The second route has a metric of two and the third route has a metric of 3. Thus, the second route is used only of the first route (which has a metric of 1) becomes unavailable. Likewise, the third route is used only if the first and second routes (which have lower metrics) are both unavailable.

For complete syntax information, refer to "Configuring a static IP route" on page 716.

Configuring standard static IP routes and interface or null static routes to the same destination

You can configure a null0 or interface-based static route to a destination and also configure a normal static route to the same destination, so long as the route metrics are different.

When the PowerConnect has multiple routes to the same destination, the PowerConnect always prefers the route with the lowest metric. Generally, when you configure a static route to a destination network, you assign the route a low metric so that the PowerConnect prefers the static route over other routes to the destination.

This feature is especially useful for the following configurations. These are not the only allowed configurations but they are typical uses of this enhancement:

- When you want to ensure that if a given destination network is unavailable, the PowerConnect drops (forwards to the null interface) traffic for that network instead of using alternate paths to route the traffic. In this case, assign the normal static route to the destination network a lower metric than the null route.
- When you want to use a specific interface by default to route traffic to a given destination network, but want to allow the PowerConnect to use other interfaces to reach the destination network if the path that uses the default interface becomes unavailable. In this case, give the interface route a lower metric than the normal static route.

NOTE
You cannot add a null or interface-based static route to a network if there is already a static route of any type with the same metric you specify for the null or interface-based route.

Figure 129 shows an example of two static routes configured for the same destination network. One of the routes is a standard static route and has a metric of 1. The other static route is a null route and has a higher metric than the standard static route. The PowerConnect always prefers the static route with the lower metric. In this example, the PowerConnect always uses the standard static route for traffic to destination network 192.168.7.0/24, unless that route becomes unavailable, in which case the PowerConnect sends traffic to the null route instead.

FIGURE 129 Standard and null static routes to the same destination network

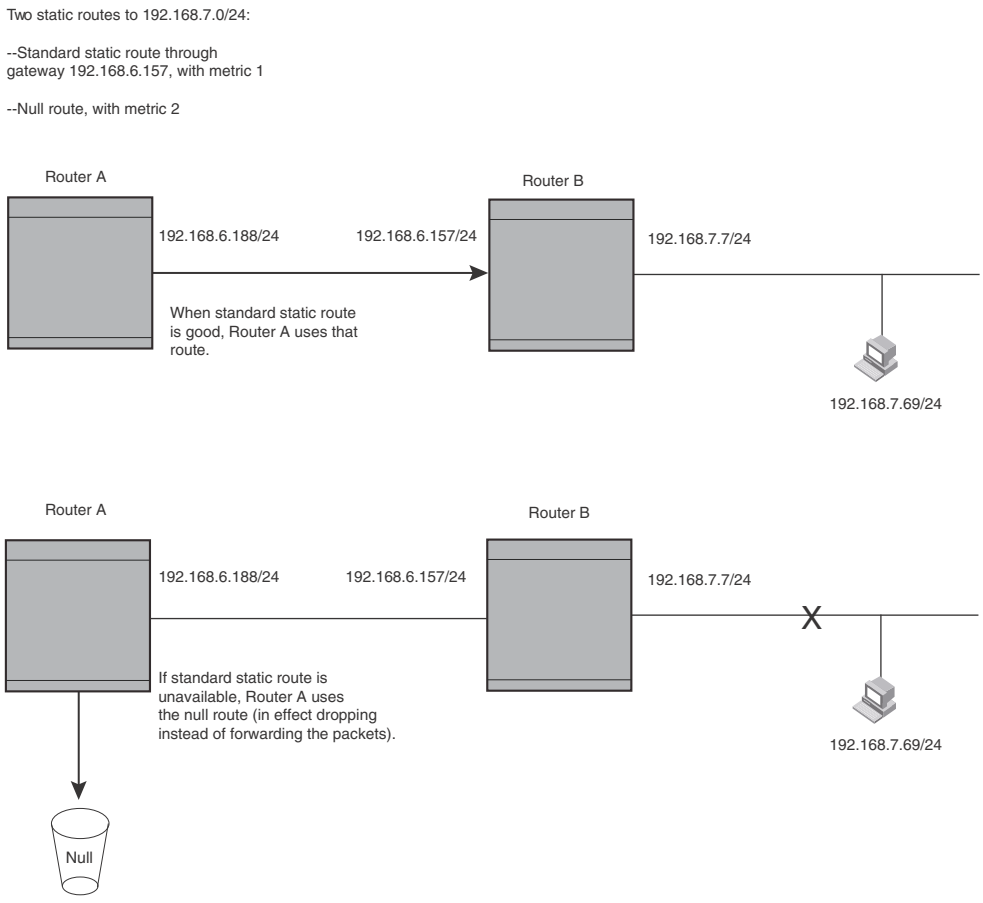
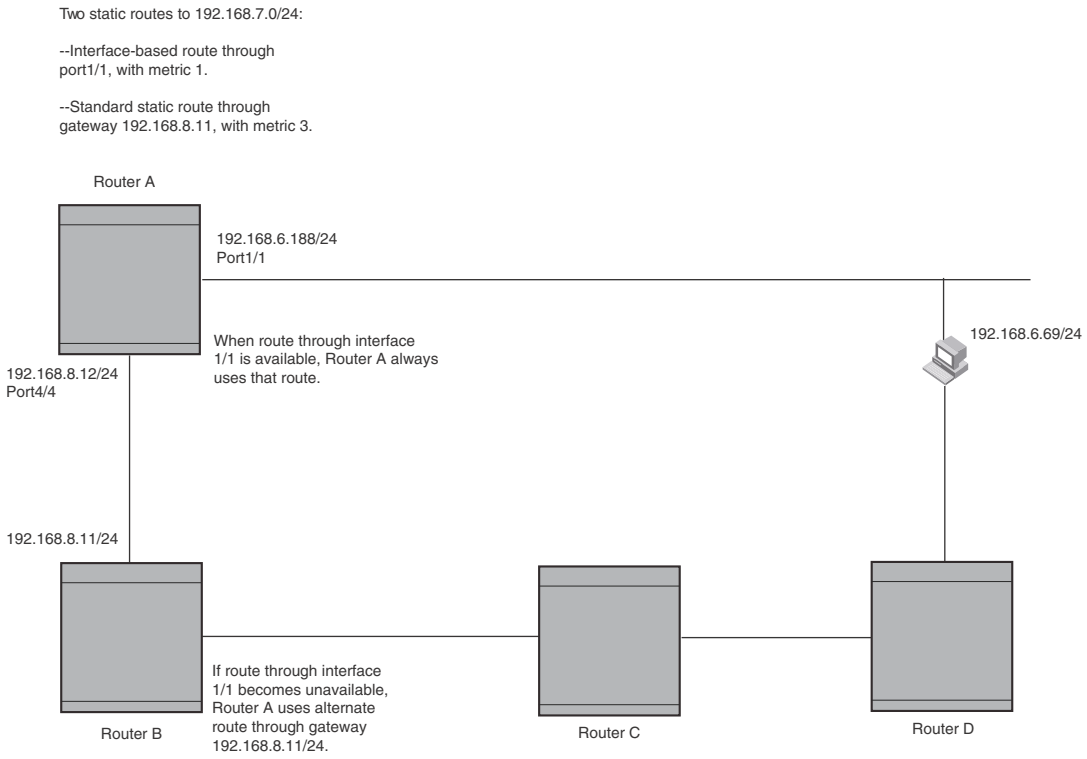


Figure 130 shows another example of two static routes. A standard static route and an interface-based static route are configured for destination network 192.168.6.0/24. The interface-based static route has a lower metric than the standard static route. As a result, the PowerConnect always prefers the interface-based route when the route is available. However, if the interface-based route becomes unavailable, the PowerConnect still forwards the traffic toward the destination using an alternate route through gateway 192.168.8.11/24.

FIGURE 130 Standard and interface routes to the same destination network



To configure a standard static IP route and a null route to the same network as shown in Figure 129 on page 723, enter commands such as the following.

```
NetIron(config)# ip route 192.168.7.0/24 192.168.6.157/24 1
NetIron(config)# ip route 192.168.7.0/24 null0 3
```

The first command configures a standard static route, which includes specification of the next-hop gateway. The command also gives the standard static route a metric of 1, which causes the PowerConnect to always prefer this route when the route is available.

The second command configures another static route for the same destination network, but the second route is a null route. The metric for the null route is 3, which is higher than the metric for the standard static route. If the standard static route is unavailable, the software uses the null route.

For complete syntax information, refer torefer to “Configuring a static IP route” on page 716.

To configure a standard static route and an interface-based route to the same destination, enter commands such as the following.

```
NetIron(config)# ip route 192.168.6.0/24 ethernet 1/1 1
NetIron(config)# ip route 192.168.6.0/24 192.168.8.11/24 3
```

The first command configured an interface-based static route through Ethernet port 1/1. The command assigns a metric of 1 to this route, causing the PowerConnect to always prefer this route when it is available. If the route becomes unavailable, the PowerConnect uses an alternate route through the next-hop gateway 192.168.8.11/24.

Static route configuration

The following enhancements to static route configuration have been added:

- "Static route tagging" on page 725
- "Static route next hop resolution" on page 725
- "Static route recursive lookup" on page 726
- "Static route resolve by default route" on page 726

Static route tagging

Static routes can be configured with a tag value, which can be used to color routes and filter routes during a redistribution process. When tagged static routes are redistributed to OSPF or to a protocol that can carry tag information, they are redistributed with their tag values.

To add a tag value to a static route, enter commands such as the following.

```
NetIron(config)# ip route 192.122.12.1 255.255.255.0 192.122.1.1 tag 20
```

Syntax: [no] ip route <*dest-ip-addr*> <*dest-mask*> | <*dest-ip-addr*>/<*dest-mask*> <*next-hop-ip-address*> tag <*value*>

The <*dest-ip-addr*> is the route's destination. The <*dest-mask*> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24. You can enter multiple static routes for the same destination for load balancing or redundancy.

The <*next-hop-ip-address*> is the IP address of the next-hop router (gateway) for the route. In addition, the <*next-hop-ip-address*> can also be a virtual routing interface (for example. ve 100), or a physical port (for example, ethernet 1/1) that is connected to the next hop router.

Enter 0 – 4294967295 for **tag** <*value*>. The default is 0, meaning no tag.

Static route next hop resolution

This feature of the Multi-Service IronWare software enables the PowerConnect to use routes from a specified protocol to resolve a configured static route. By default this is disabled.

To configure static route next hop resolution with OSPF routes, use the following command.

```
NetIron(config)# ip route next-hop ospf
```

Syntax: [no] ip route next-hop [bgp | isis | ospf | rip]

NOTE
This command can be independently applied on a per-VRF basis.

This command causes the resolution of static route next hop using routes learned from one of the following protocols:

- bgp – both iBGP and eBGP routes are used to resolve static routes.
- isis
- ospf
- rip

NOTE
Connected routes are always used to resolve static routes.

Static route recursive lookup

This feature of the Multi-Service IronWare software enables the PowerConnect to use static routes to resolve another static route. The recursive static route nexthop lookup level can be configured. By default, this feature is disabled.

To configure static route next hop recursive lookup by other static routes, use the following command.

```
NetIron(config)# ip route next-hop-recursion 5
```

Syntax: [no] ip route next-hop-recursion <*level*>

The <*level*> available specifies the numbers of level of recursion allowed. Acceptable values are 1-10. The default value is 3.

NOTE
This command can be independently applied on a per-VRF basis.

Static route resolve by default route

This feature of the Multi-Service IronWare software enables the PowerConnect to use the default route (0.0.0.0/0) to resolve a static route. By default, this feature is disabled.

Use the following command to configure static route resolve by default route.

```
NetIron(config)# ip route next-hop-enable-default
```

Syntax: [no] ip route next-hop-enable-default

NOTE
This command can be independently applied on a per-VRF basis.

NOTE
This command works independently with the **ip route next-hop-recursion** and **ip route next-hop** commands. If the default route is a protocol route, that protocol needs to be enabled to resolve static routes using the **ip route next-hop [protocol-name]** command in order for static routes to resolve by this default route. If the default route itself is a static route, you must configure the **ip route next-hop-recursion** command to resolve other static routes by this default route.

Static route to an LSP tunnel interface

This feature allows you to set the next hop for a static route to the egress router of an LSP tunnel if the destination route is contained in the MPLS routing table. In this configuration, the static route is updated with the LSP routes and reverts to its original next hop outgoing interface when this feature is disabled or when the LSP goes down. This route can be used for the default route.

To enable the static route to an LSP tunnel interface feature, use the following command.

```
NetIron(config)# ip route next-hop-enable-mpls
```

Syntax: [no] ip route next-hop-enable-mpls

The static route can then be directed to the IP address of the egress router of the LSP. In the following example, a static route is configured to network 10.10.10.0/24 through 11.11.11.1, which is the IP address of the egress router of an LSP tunnel.

```
NetIron(config)# ip route 10.10.10.0/24 11.11.11.1
```

As previously stated, this feature works only if a route to the destination network is contained in the MPLS routing table. To verify that it is, you can use the **show ip route** command, as shown in the following example.

```
NetIron# show ip route
Total number of IP routes: 6
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
        Destination        Gateway         Port           Cost           Type
1       10.47.6.0/24       DIRECT          mgmt 1         0/0            D
2       11.11.11.11/32     DIRECT          loopback 1     0/0            D
3       12.12.12.12/32     40.1.0.1        eth 5/1        110/3          O
        12.12.12.12/32     12.12.12.12     lsp delll   110/3          O
                                           onglspfoundr
                                           ylonglspfoun
                                           drylonglspfo
                                           undrylonglsp
                                           fou
        12.12.12.12/32     12.12.12.12     lsp t11        110/3          O
4       13.13.13.13/32     40.1.0.1        eth 5/1        110/2          O
5       40.1.0.0/24        DIRECT          eth 5/1        0/0            D
6       50.1.0.0/24        40.1.0.1        eth 5/1        110/2          O
PE1-#
```

As shown in the example, route 2 has a destination network of 10.10.10.0/24 through the gateway at IP address 11.11.11.1 which is on LSP 12.

NOTE

The show commands are enhanced to include the full LSP name. Previously, the LSP name was truncated because it exceeded the character length. Now, the LSP name is text wrapped to display the full name. For example, see show ip route above.

To verify that an LSP is up and that MPLS has a route to it, you can use the **show mpls lsp** and **show mpls route** commands as shown in the following examples.

```
NetIron# show mpls lsp
Note: LSPs marked with * are taking a Secondary Path
                                Admin  Oper  Tunnel   Up/Dn  Retry   Active
Name            To              State State Intf     Times  No.     Path
t6              12.12.12.13     UP    DOWN    --      0      292     --
t5              12.12.12.12     UP    UP     tnl3     1      0       --
t2              12.12.12.12     UP    UP     tnl1     1      0       --
t7              12.12.12.12     UP    UP     tnl5     1      0       one
powerconnectlongl   12.12.12.12     UP    UP     tnl9     1      0       --
spverylongls
pveryverylon
glsp
t4              12.12.12.12     UP    UP     tnl2     1      0        --
t1              12.12.12.12     UP    UP     tnl0     1      0         --



NetIron(config-mpls-if-e100-1/3)#show mpls route
Total number of MPLS tunnel routes: 3
R:RSVP L:LDP S:Static O:Others
      Destination          Gateway           Tnnl    Port  Label  Sig Cost Use
1     2.2.2.2/32           2.2.2.2           tnl8    1/2   3      L   0    0
2     3.3.3.3/32           2.2.2.2           tnl9    1/1   1025   L   0    0
3     3.3.3.3/32           3.3.3.3           tnl1    1/2   1027   R   0    0
```

NOTE
The show commands have been enhanced to include the full MPLS tunnel name. Previously, the MPLS tunnel name was truncated because it exceeded the character length. Now, the MPLS tunnel name is text-wrapped to display the full name. For an illustration, see the output of the **show mpls lsp** and **show mpls route** commands in the preceding examples.

Naming a static IP route

You can assign a name to a static IP route. A static IP route name serves as a description of the route. The name can be used to more readily reference or identify the associated static route.

NOTE
The static route name is an optional feature. It does not affect the selection of static routes.

The Dell device does not check for the uniqueness of names assigned to static routes. Static routes that have the same or different next hop(s) can have the same or different name(s). Due to this, the same name can be assigned to multiple static routes to group them. The name is then used to reference or identify a group of static routes.

NOTE
This feature is supported on standard static IP routes and static IP routes between VRFs (both default and non-default).

The option to assign a name to a static route is displayed after you select either an outgoing interface type or configure the next hop address.

To assign a name to a static route, enter commands such as the following.

```
NetIron(config)# ip route 12.22.22.22 255.255.255.255 eth 1/1 name abc
```

OR

```
NetIron(config)# ip route 12.22.22.22 255.255.255.255 12.1.1.1 name abc
```

Syntax: [no] **ip route** <*dest-ip-addr*> <*dest-mask*> | <*dest-ip-addr*>/<*mask-bits*> <*next-hop-ip-addr*>| **ethernet** <*slot/port*>| **pos** <*slot/port*> | **ve** <num> [<metric>] [**tag** <num>] [**distance** <num>] [**name** <*string*>]

Enter the static route name for **name** <*string*>. The maximum length of the name is 128 bytes.

The output of the **show** commands displays the name of a static IP route if there is one assigned.

Show run displays the entire name of the static IP route. The **show ip static route** command displays an asterisk (*) after the first twelve characters if the assigned name is thirteen characters or more. The **show ipv6 static route** command displays an asterisk after the first two characters if the assigned name is three characters or more.

When displayed in **show run**, a static route name with a space in the name will appear within quotation marks (for example, “brcd route”).

Refer to “Displaying the IP static route table for a VRF” on page 1683.

Changing the name of a static IP route

To change the name of a static IP route, enter the static route as configured. Proceed to enter the new name instead of the previous name. See the example below.

Static IP route with the original name “abc”:

```
NetIron(config)# ip route 12.22.22.22 255.255.255.255 12.1.1.1 name abc
```

Change the name of “abc” to “xyz”:

```
NetIron(config)# ip route 12.22.22.22 255.255.255.255 12.1.1.1 name xyz
```

In this example, “xyz” is the set as the new name of the static IP route.

Deleting the name of a static IP route

To delete the name of a static IP route, use the **no** command. See the example below.

Static IP route with the name “xyz”:

```
NetIron(config)# ip route 12.22.22.22 255.255.255.255 12.1.1.1 name xyz
```

To remove the name “xyz” from the static IP route, specify both “name” and the string, in this case “xyz”.

```
NetIron(config)#no ip route 12.22.22.22 255.255.255.255 12.1.1.1 name xyz
```

The static route no longer has a name assigned to it.

Configuring a default network route

The PowerConnect enables you to specify a candidate default route without the need to specify the next hop gateway. If the IP route table does not contain an explicit default route (for example, 0.0.0.0/0) or propagate an explicit default route through routing protocols, the software can use the default network route as a default route instead.

When the software uses the default network route, it also uses the default network route's next hop gateway as the gateway of last resort.

This feature is especially useful in environments where network topology changes can make the next hop gateway unreachable. This feature allows the PowerConnect to perform default routing even if the default network route's default gateway changes.

The feature thus differs from standard default routes. When you configure a standard default route, you also specify the next hop gateway. If a topology change makes the gateway unreachable, the default route becomes unusable.

For example, if you configure 10.10.10.0/24 as a candidate default network route, if the IP route table does not contain an explicit default route (0.0.0.0/0), the software uses the default network route and automatically uses that route's next hop gateway as the default gateway. If a topology change occurs and as a result the default network route's next hop gateway changes, the software can still use the default network route.

If you configure more than one default network route, the PowerConnect uses the following algorithm to select one of the routes.

1. Use the route with the lowest administrative distance.
2. If the administrative distances are equal:
    - Are the routes from different routing protocols (RIP, OSPF, or BGP4)? If so, use the route with the lowest IP address.
    - If the routes are from the same routing protocol, use the route with the best metric. The meaning of “best” metric depends on the routing protocol:
    - **RIP** – The metric is the number of hops (additional routers) to the destination. The best route is the route with the fewest hops.
    - **OSPF** – The metric is the path cost associated with the route. The path cost does not indicate the number of hops but is instead a numeric value associated with each route. The best route is the route with the lowest path cost.
    - **BGP4** – The metric is the Multi-exit Discriminator (MED) associated with the route. The MED applies to routes that have multiple paths through the same AS. The best route is the route with the lowest MED.

NOTE
Currently the PowerConnect will *not* propagate a candidate default route, specified by the "**ip default-network** <>" command, into the routing protocols in spite of the **default-information-originate** command being configured under the routing protocols.

Configuring a default network route

You can configure up to four default network routes. To configure a default network route, enter commands such as the following.

```
NetIron(config)# ip default-network 209.157.22.0
NetIron(config)# write memory
```

Syntax: [no] ip default-network <*ip-addr*>

The <*ip-addr*> parameter specifies the network address.

To verify that the route is in the route table, enter the following command at any level of the CLI.

```
NetIron(config)# show ip route
Total number of IP routes: 2
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
      Destination       Gateway          Port   Cost   Type
1     209.157.20.0      0.0.0.0          lb1    1      D
2     209.157.22.0      0.0.0.0          4/11   1      *D
```

This example shows two routes. Both of the routes are directly attached, as indicated in the Type column. However, one of the routes is shown as type "*D", with an asterisk (*). The asterisk indicates that this route is a candidate default network route.

Configuring IP load sharing

The IP route table can contain more than one path to a given destination. When this occurs, the PowerConnect selects the path with the lowest cost as the path for forwarding traffic to the destination. If the IP route table contains more than one path to a destination and the paths each have the lowest cost, then the PowerConnect uses ***IP load sharing*** to select a path to the destination.[1]

IP load sharing is based on the destination address of the traffic. PowerConnect supports load sharing based on individual host addresses or on network addresses.

You can enable a PowerConnect to load balance across up to eight equal-cost paths. The default maximum number of equal-cost load sharing paths is four.

NOTE
IP load sharing is not based on source routing, only on next-hop routing.

NOTE
The term "path" refers to the next-hop router to a destination, not to the entire route to a destination. Thus, when the software compares multiple equal-cost paths, the software is comparing paths that use different next-hop routers, with equal costs, to the same destination.

In many contexts, the terms "route" and "path" mean the same thing. Most of the user documentation uses the term "route" throughout. The term "path" is used in this section to refer to an individual next-hop router to a destination, while the term "route" refers collectively to the multiple paths to the destination. Load sharing applies when the IP route table contains multiple, equal-cost paths to a destination.

NOTE
The PowerConnect also performs load sharing among the ports in aggregate links.

1. IP load sharing is also called "Equal-Cost Multi-Path (ECMP)" load sharing or just "ECMP"

How multiple equal-cost paths enter the IP route table

IP load sharing applies to equal-cost paths in the IP route table. Routes eligible for load sharing can enter the table from the following sources:

- IP static routes
- Routes learned through RIP, OSPF, and BGP4

Administrative distance

The administrative distance is a unique value associated with each type (source) of IP route. Each path has an administrative distance. It is used when evaluating multiple equal-cost paths to the same destination from different sources, such as RIP, OSPF and so on, but not used when performing IP load sharing.

The value of the administrative distance is determined by the source of the route. The PowerConnect is configured with a unique administrative distance value for each IP route source.

When the software receives paths from different sources to the same destination, the software compares their administrative distances, selects the one with the lowest distance, and puts it in the IP route table. For example, if the PowerConnect has a path learned from OSPF and a path learned from RIP for a given destination, only the path with the lower administrative distance enters the IP route table.

Here are the default administrative distances on the PowerConnect:

- Directly connected – 0 (this value is not configurable)
- Static IP route – 1 (applies to all static routes, including default routes and default network routes)
- Exterior Border Gateway Protocol (EBGP) – 20
- OSPF – 110
- RIP – 120
- Interior Gateway Protocol (IBGP) – 200
- Local BGP – 200
- Unknown – 255 (the router will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the router receives routes for the same network from OSPF and from RIP, the router will prefer the OSPF route by default.

NOTE
You can change the administrative distances individually. Refer to the configuration chapter for the route source for information.

Since the software selects only the path with the lowest administrative distance, and the administrative distance is determined by the path's source, IP load sharing does not apply to paths from different route sources. IP load sharing applies only when the IP route table contains paths from the same IP route source to the same destination.

Path cost

The cost parameter provides a basis of comparison for selecting among paths to a given destination. Each path in the IP route table has a cost. When the IP route table contains multiple paths to a destination, the PowerConnect chooses the path with the lowest cost. When the IP route table contains more than one path with the lowest cost to a destination, the PowerConnect uses IP load sharing to select one of the lowest-cost paths.

The source of a path's cost value depends on the source of the path:

- **IP static route** – The value you assign to the metric parameter when you configure the route. The default metric is 1. Refer to "Configuring load balancing and redundancy using multiple static routes to the same destination" on page 721.
- **RIP** – The number of next-hop routers to the destination.
- **OSPF** – The Path Cost associated with the path. The paths can come from any combination of inter-area, intra-area, and external Link State Advertisements (LSAs).
- **BGP4** – The path's Multi-Exit Discriminator (MED) value.

NOTE
If the path is redistributed between two or more of the above sources before entering the IP route table, the cost can increase during the redistribution due to settings in redistribution filters.

Static route, OSPF, and BGP4 load sharing

IP load sharing and load sharing for static routes, OSPF routes, and BGP4 routes are individually configured. Multiple equal-cost paths for a destination can enter the IP route table only if the source of the paths is configured to support multiple equal-cost paths. For example, if BGP4 allows only one path with a given cost for a given destination, the BGP4 route table cannot contain equal-cost paths to the destination. Consequently, the IP route table will not receive multiple equal-cost paths from BGP4.

Table 113 lists the default and configurable maximum numbers of paths for each IP route source that can provide equal-cost paths to the IP route table. The table also lists where to find configuration information for the route source's load sharing parameters.

The load sharing state for all the route sources is based on the state of IP load sharing. Since IP load sharing is enabled by default on the PowerConnect, load sharing for static IP routes, RIP routes, OSPF routes, and BGP4 routes also is enabled by default.

TABLE 113 Default load sharing parameters for route sources

Route source	Default maximum number of paths	Maximum number of paths	See...
Static IP route	4 **NOTE:** This value depends on the value for IP load sharing, and is not separately configurable.	8 **NOTE:** This value depends on the value for IP load sharing, and is not separately configurable.	page 737
RIP	4 **NOTE:** This value depends on the value for IP load sharing, and is not separately configurable.	8 **NOTE:** This value depends on the value for IP load sharing, and is not separately configurable.	page 737

TABLE 113 Default load sharing parameters for route sources (Continued)

Route source	Default maximum number of paths	Maximum number of paths	See...
OSPF	4	8	page 737
BGP4	1	4	page 1037

Options for IP load sharing and LAGs

The following options have been added to refine the hash calculations used for IP load sharing and LAGs. These include the following:

- **Speculate UDP or TCP Headers** – This option is applied to ECMP and LAG index hash calculations.
- **Mask Layer-3 and Layer-4 Information** – This option is applied to ECMP and LAG index hash calculations.
- **Mask Layer-2 Information** – This option is applied to ECMP and LAG index hash calculations.
- **Diversification** – This option is applied to ECMP and LAG index hash calculations.
- **Hash Rotate** – This option is applied to ECMP hash calculations and to LAG index calculations.

Speculate UDP or TCP packet headers

With this option set, the packet headers following IPv4 headers are used for the ECMP and LAG index hash calculations even if the packet is not a TCP or UDP packet. If the packet is a non-fragmented. no-IP options, true TCP or UDP packet, the TCP or UDP ports will be used for hash calculations unless the **load-balance mask ip** or **load-balance mask ipv6** commands are used. This behavior is off by default and can be enabled using the following command.

```
NetIron(config)# load-balance force-l4-hashing all
```

Syntax: **[no] load-balance force-l4-hashing [all |** <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

The **all** option applies the command to all ports within the router.

Specifying a slot number using the <*slot-number*> variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the <*slot-number*> and <*np-id*> variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE
Problems can occur with the **Ping** and **Traceroute** functions when this option is enabled.

Masking layer-3 and layer-4 information

With this option set, the following values can be masked during ECMP and LAG index hash calculations: TCP or UDP source and destination port information, source and destination IP address, IPv4 protocol ID, and IPv6 next header. When used with the **load-balance force-l4-hashing** command, this command takes precedence. This option can be set using the following commands.

```
NetIron(config)# load-balance mask ip src-l4-port all
```

Syntax: **[no] load-balance mask ip [src-l4-port | dst-l4-port | src-ip | dst-ip | protocol] [all |** <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

Use the **src-l4-port** option when you want to mask the Layer-4 source port.

Use the **dst-l4-port** option when you want to mask the Layer-4 destination port.

Use the **src-ip** option when you want to mask the IP source address.

Use the **dst-ip** option when you want to mask the IP destination address.

Use the **protocol** option when you want to mask the IPv4 protocol ID.

The **all** option applies the command to all ports within the router.

Specifying a slot number using the <*slot-number*> variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the <*slot-number*> and <*np-id*> variables limits the command to the ports supported by the specified network processor on the specified interface module.

This command can be applied for IPv6 using the following command.

```
NetIron(config)# load-balance mask ipv6 src-l4-port all
```

Syntax: [no] **load-balance mask ipv6 [src-l4-port | dst-l4-port | src-ip | dst-ip | next-hdr] [all |** <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

Except for the **next-hdr** option, the command options are the same as for the **load-balance mask ip** command as described previously. The **next-hdr** option is described in the following:

Use the **next-hdr** option when you want to mask the IPv6 next header.

These commands are disabled by default.

Masking layer-2 information

With the **load-balance mask ethernet** command set, the following Layer-2 values can be masked during ECMP and LAG index hash calculations: source and destination MAC address, VLAN, Ethertype, and Inner VLAN. To mask Layer-2 information, use the **load-balance mask ethernet** command, as shown in the following.

```
NetIron(config)# load-balance mask ethernet sa-mac all
```

Syntax: [no] **load-balance mask ethernet [sa-mac | da-mac | vlan | etype | inner-vlan] [all |** <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

Use the **sa-mac** option when you want to mask the source MAC address.

Use the **da-mac** option when you want to mask the destination MAC address.

Use the **vlan** option when you want to mask the VLAN ID.

Use the **etype** option when you want to mask the Ethertype

Use the **inner-vlan** option when you want to mask the inner VLAN ID.

The **all** option applies the command to all ports within the router.

Specifying a slot number using the <*slot-number*> variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the <*slot-number*> and <*np-id*> variables limits the command to the ports supported by the specified network processor on the specified interface module.

Masking MPLS information

With the **loadbalance mask mpls** command set, MPLS Labels 0, 1 or 2 can be masked during ECMP and LAG index hash calculation. To mask MPLS information, use the **load-balance mask mpls** command, as shown in the following.

```
NetIron(config)# load-balance mask mpls label0 all
```

Syntax: **[no] load-balance mask mpls [label0 | label1 | label2] [all |** <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

Use the **label0** option when you want to mask MPLS Label 0 which is the outer-most MLS label in a packet.

Use the **label1** option when you want to mask MPLS Label 1 which is the next outer-most MLS label in a packet from MPLS Label 2.

Use the **label2** option when you want to mask MPLS Label 2 which is the inner-most MLS label in a packet.

The **all** option applies the command to all ports within the router.

Specifying a slot number using the <*slot-number*> variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the <*slot-number*> and <*np-id*> variables limits the command to the ports supported by the specified network processor on the specified interface module.

Hash diversification for LAGs and IP load balancing

In a multi-stage network a traffic flow will normally use the same LAG port or same path (for IP load balancing) at each stage. The Hash Diversification feature works within an earlier stage of the hash calculation than the hash rotate feature. Using the **load-balance hash-diversify** command, you can provide a unique hash diversify value to a router, or a sub-set of ports on a router. This unique value is used in calculation of the ECMP and LAG index hash. Consequently, instead of a traffic flow always following the same port group or path, it will be distributed over different LAG or ECMP members. To apply hash diversification, use the following command.

```
NetIron(config)# load-balance hash-diversify random all
```

Syntax: **[no] load-balance hash-diversify [** <*number*> **| random | slot>] [all |** <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

You can set the unique hash diversify value using one of the following options:

The <*number*> option allows you to specify a value from 0 - 255.

The **random** option directs the CPU to generate a random number for each packet processor and program it as the hash diversification value.

The **slot** option specifies the slot ID as the has diversification number.

The default value for the diversification number is 0 and the **no** version of the command resets the value to 0 regardless of any value previously set. Also, the most recent command added overrides any previous instances of the command. For example, if the **random** option is entered first and is then followed by the **slot** option, the value of the slot ID for the specified slot will be used.

The **all** option applies the command to all ports within the router.

Specifying a slot number using the <*slot-number*> variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the <*slot-number*> and <*np-id*> variables limits the command to the ports supported by the specified network processor on the specified interface module.

This option can also be used in a multi-stage network to avoid the same traffic flow to always use one path of an ECMP or the same LAG member index at each stage. Using the hash rotate function the same set of traffic flows forwarded out of one LAG member or ECMP path to the next router can be distributed across different paths of the LAG member or ECMP path to the next router.

Hash rotate for LAGs and IP load balancing

The hash rotate function provides another option (in addition to hash diversification) for diversifying traffic flow in a multi-stage network. Using this feature, the ECMP hash index can be rotated by a specified number of bits after it has been calculated. This allows path selection within IP load balancing to be more diverse.

To configure hash rotate to LAG index calculations, enter a command such as the following.

```
NetIron(config)# load-balance hash-rotate 3 all
```

Syntax: **[no] load-balance hash-rotate** <*rotate-number*> **[all |** <*slot-number*> **|** <*slot-number*> <*np-id*> **]**

The <*rotate-number*> value specifies number of bits between 0 and 7 that you want to rotate the ECMP hash index value.

The **all** option applies the command to all ports within the router.

Specifying a slot number using the <*slot-number*> variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the <*slot-number*> and <*np-id*> variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE
The hash diversification and hash rotate features can be applied seperately or together. Depending on your network configuration, either or both of these features may need to be configured.

How IP load sharing works

On the PowerConnect, IP load sharing is done by the hardware. If there is more than one path to a given destination, a hash is calculated based on the source MAC address, destination MAC address, source IP address, destination IP address, VLAN-ID (if applicable), IPv4 protcol number, IPv6 next header and TCP/UDP source port and destination port if the packet is also a TCP/UDP packet. This hash is used to select one of the paths.

Changing the maximum number of load sharing paths

By default, IP load sharing allows IP traffic to be balanced across up to four equal path. You can change the maximum number of paths that the PowerConnect supports to a value of 2 – 8.

For optimal results, set the maximum number of paths to a value equal to or greater than the maximum number of equal-cost paths that your network typically contains. For example, if the PowerConnect has six next-hop routers, set the maximum paths value to six.

NOTE
If the setting for the maximum number of paths is lower than the actual number of equal-cost paths, the software does not use all the paths for load sharing.

To change the number of paths, enter a command such as the following.

```
NetIron(config)# ip load-sharing 8
```

Syntax: [no] ip load-sharing [<*number*>**]**

Enter a value from 2 – 8 for <*number*> to set the maximum number of paths.

NOTE
A new command was introduced (maximum-paths use-load-sharing) within the BGP configuration that allows support for BGP routes in IP Load Sharing while BGP Multipath Load sharing is not enable.

Response to path state changes

If one of the load-balanced paths becomes unavailable, the IP route table in hardware is modified to stop using the unavailable path. The traffic is load balanced between the available paths using the same hashing mechanism described above. (Refer to "How IP load sharing works" on page 737.)

Configuring IRDP

The PowerConnect uses ICMP Router Discovery Protocol (IRDP) to advertise the IP addresses of its router interfaces to directly attached hosts. IRDP is disabled by default. You can enable it globally or on individual ports.

Consider the following when you enable or disable IRDP globally:

- If you enable IRDP globally, all ports use the default values for the IRDP parameters.
- If you leave IRDP disabled globally but enable it on individual ports, you also can configure the IRDP parameters on an individual port basis.

  NOTE
  You can configure IRDP parameters only an individual port basis. To do so, IRDP must be disabled globally and enabled only on individual ports. You cannot configure IRDP parameters if the feature is globally enabled.

When IRDP is enabled, the PowerConnect periodically sends Router Advertisement messages out the IP interfaces on which the feature is enabled. The messages advertise the PowerConnect's IP addresses to directly attached hosts who listen for the messages. In addition, hosts can be configured to query the PowerConnect for the information by sending Router Solicitation messages.

Some types of hosts use the Router Solicitation messages to discover their default gateway. When IRDP is enabled, the PowerConnect responds to the Router Solicitation messages. Some clients interpret this response to mean that the PowerConnect is the default gateway. If another router is actually the default gateway for these clients, leave IRDP disabled on the PowerConnect.

IRDP uses the following parameters. If you enable IRDP on individual ports rather than globally, you can configure these parameters on an individual port basis. The IRDP parameters are as follows:

- **Packet type** – The PowerConnect can send Router Advertisement messages as IP broadcasts or as IP multicasts addressed to IP multicast group 224.0.0.1. The packet type is IP broadcast.
- **Maximum message interval and minimum message interval** – When IRDP is enabled, the PowerConnect sends the Router Advertisement messages every 450 – 600 seconds by default. The time within this interval that the PowerConnect selects is random for each message and is not affected by traffic loads or other network factors. The random interval minimizes the probability that a host will receive Router Advertisement messages from other routers at the same time. The interval on each IRDP-enabled PowerConnect interface is independent of the interval on other IRDP-enabled interfaces. The default maximum message interval is 600 seconds. The default minimum message interval is 450 seconds.
- **Hold time** – Each Router Advertisement message contains a hold time value. This value specifies the maximum amount of time the host should consider an advertisement to be valid until a newer advertisement arrives. When a new advertisement arrives, the hold time is reset. The hold time is always longer than the maximum advertisement interval. Therefore, if the hold time for an advertisement expires, the host can reasonably conclude that the router interface that sent the advertisement is no longer available. The default hold time is three times the maximum message interval.
- **Preference** – If a host receives multiple Router Advertisement messages from different routers, the host selects the router that sent the message with the highest preference as the default gateway. The preference can be a number from 4294967296 to 4294967295. The default is 0.

Enabling IRDP globally

To globally enable IRDP, enter the following command.

```
NetIron(config)# ip irdp
```

This command enables IRDP on the IP interfaces on all ports. Each port uses the default values for the IRDP parameters. The parameters are not configurable when IRDP is globally enabled.

Enabling IRDP on an individual port

To enable IRDP on an individual interface and change IRDP parameters, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/3
NetIron(config-if-e10000-1/3)# ip irdp maxadvertinterval 400
```

This example shows how to enable IRDP on a specific port and change the maximum advertisement interval for Router Advertisement messages to 400 seconds.

NOTE
To enable IRDP on individual ports, you must leave the feature globally disabled.

Syntax: **[no] ip irdp [broadcast | multicast] [holdtime** <*seconds*>**] [maxadvertinterval** <*seconds*>**] [minadvertinterval** <*seconds*>**] [preference** <*number*>**]**

The **broadcast | multicast** parameter specifies the packet type the PowerConnect uses to send Router Advertisement.

- **broadcast** – The PowerConnect sends Router Advertisement as IP broadcasts. This is the default.

- **multicast** – The PowerConnect sends Router Advertisement as multicast packets addressed to IP multicast group 224.0.0.1.

The **holdtime** <*seconds*> parameter specifies how long a host that receives a Router Advertisement from the PowerConnect should consider the advertisement to be valid. When a host receives a new Router Advertisement message from the PowerConnect, the host resets the hold time for the PowerConnect to the hold time specified in the new advertisement. If the hold time of an advertisement expires, the host discards the advertisement, concluding that the router interface that sent the advertisement is no longer available. The value must be greater than the value of the **maxadvertinterval** parameter and cannot be greater than 9000. The default is three times the value of the **maxadvertinterval** parameter.

The **maxadvertinterval** parameter specifies the maximum amount of time the PowerConnect waits between sending Router Advertisements. You can specify a value from 1 to the current value of the **holdtime** parameter. The default is 600 seconds.

The **minadvertinterval** parameter specifies the minimum amount of time the PowerConnect can wait between sending Router Advertisements. The default is three-fourths (0.75) the value of the **maxadvertinterval** parameter. If you change the **maxadvertinterval** parameter, the software automatically adjusts the **minadvertinterval** parameter to be three-fourths the new value of the **maxadvertinterval** parameter. If you want to override the automatically configured value, you can specify an interval from 1 to the current value of the **maxadvertinterval** parameter.

The **preference** <*number*> parameter specifies the IRDP preference level of the PowerConnect. If a host receives Router Advertisements from multiple routers, the host selects the router interface that sent the message with the highest interval as the host's default gateway. The valid range is 4294967296 to 4294967295. The default is 0.

Configuring UDP broadcast and IP helper parameters

Some applications rely on client requests sent as limited IP broadcasts addressed to the UDP's application port. If a server for the application receives such a broadcast, the server can reply to the client. Routers do not forward subnet directed broadcasts, so the client and server must be on the same network for the broadcast to reach the server. If the client and server are on different networks (on opposite sides of a router), the client's request cannot reach the server.

To configure the PowerConnect to forward clients' requests to UDP application servers:

- Enable forwarding support for the UDP application port, if forwarding support is not already enabled.
- Configure a helper adders on the interface connected to the clients. Specify the helper address to be the IP address of the application server or the subnet directed broadcast address for the IP subnet the server is in. A helper address is associated with a specific interface and applies only to client requests received on that interface. The PowerConnect forwards client requests for any of the application ports the PowerConnect is enabled to forward to the helper address.

Forwarding support for the following application ports is enabled by default:

- bootps (port 67)
- dns (port 53)
- tftp (port 69)
- time (port 37)
- netbios-ns (port 137)

- netbios-dgm (port 138)
- tacacs (port 65)

NOTE
The application names are the names for these applications that the PowerConnect recognizes, and might not match the names for these applications on some third-party devices. The numbers listed in parentheses are the UDP port numbers for the applications. The numbers come from RFC 1340.

NOTE
As shown above, forwarding support for BootP or DHCP is enabled by default. If you are configuring the PowerConnect to forward BootP or DHCP requests, refer to "Configuring BootP or DHCP forwarding parameters" on page 742.

You can enable forwarding for other applications by specifying the application port number.

You also can disable forwarding for an application.

NOTE
If you disable forwarding for a UDP application, forwarding of client requests received as broadcasts to helper addresses is disabled. Disabling forwarding of an application does not disable other support for the application. For example, if you disable forwarding of Telnet requests to helper addresses, other Telnet support on the PowerConnect is not also disabled.

Enabling forwarding for a UDP application

If you want the PowerConnect to forward client requests for UDP applications that the PowerConnect does not forward by default, you can enable forwarding support for the port. To enable forwarding support for a UDP application, use either of the following methods. You also can disable forwarding for an application using these methods.

NOTE
You also must configure a helper address on the interface that is connected to the clients for the application. The PowerConnect cannot forward the requests unless you configure the helper address. Refer to "Configuring an IP helper address" on page 743.

To enable the forwarding of SNMP trap broadcasts, enter the following command.

```
NetIron(config)# ip forward-protocol udp snmp-trap
```

Syntax: [no] ip forward-protocol udp <*udp-port-name*> | <*udp-port-num*>

The <*udp-port-name*> parameter can have one of the following values. For reference, the corresponding port numbers from RFC 1340 are shown in parentheses. If you specify an application name, enter the name only, not the parentheses or the port number shown here:

- bootpc (port 68)
- bootps (port 67)
- discard (port 9)
- dns (port 53)
- dnsix (port 90)
- echo (port 7)
- mobile-ip (port 434)

- netbios-dgm (port 138)
- netbios-ns (port 137)
- ntp (port 123)
- tacacs (port 65)
- talk (port 517)
- time (port 37)
- tftp (port 69)

In addition, you can specify any UDP application by using the application's UDP port number.

The <*udp-port-num*> parameter specifies the UDP application port number. If the application you want to enable is not listed above, enter the application port number. You also can list the port number for any of the applications listed above.

To disable forwarding for an application, enter a command such as the following.

```
NetIron(config)# no ip forward-protocol udp snmp
```

This command disables forwarding of SNMP requests to the helper addresses configured on PowerConnect interfaces.

Configuring an IP helper address

To forward a client's broadcast request for a UDP application when the client and server are on different networks, you must configure a helper address on the interface connected to the client. Specify the server's IP address or the subnet directed broadcast address of the IP subnet the server is in as the helper address.

You can configure up to 16 helper addresses on each interface. You can configure a helper address on an Ethernet port or a virtual interface.

To configure a helper address on interface 2 on chassis module 1, enter the following commands.

```
NetIron(config)# interface e 1/2
NetIron(config-if-e1000-1/2)# ip helper-address 207.95.7.6
```

The commands in this example change the CLI to the configuration level for port 1/2, then add a helper address for server 207.95.7.6 to the port. If the port receives a client request for any of the applications that the PowerConnect is enabled to forward, the PowerConnect forwards the client's request to the server.

Syntax: [no] ip helper-address <*ip-addr*>

The <*ip-addr*> command specifies the server's IP address or the subnet directed broadcast address of the IP subnet the server is in.

Configuring BootP or DHCP forwarding parameters

A host on an IP network can use BootP or DHCP to obtain its IP address from a BootP or DHCP server. To obtain the address, the client sends a BootP or DHCP request. The request is a subnet directed broadcast and is addressed to UDP port 67. A limited IP broadcast is addressed to IP address 255.255.255.255 and is not forwarded by the PowerConnect or other IP routers.

When the BootP or DHCP client and server are on the same network, the server receives the broadcast request and replies to the client. However, when the client and server are on different networks, the server does not receive the client's request, because the PowerConnect does not forward the request.

You can configure the PowerConnect to forward BootP or DHCP requests. To do so, configure a helper address on the interface that receives the client requests, and specify the BootP or DHCP server's IP address as the address you are helping the BootP or DHCP requests to reach. Instead of the server's IP address, you can specify the subnet directed broadcast address of the IP subnet the server is in.

NOTE
The IP subnet configured on the port which is directly connected to the device sending a BootP or DHCP request, does not have to match the subnet of the IP address given by the DHCP server.

BootP or DHCP forwarding parameters

The following parameters control the PowerConnect's forwarding of BootP or DHCP requests:

- **Helper address** – The BootP or DHCP server's IP address. You must configure the helper address on the interface that receives the BootP or DHCP requests from the client. The PowerConnect cannot forward a request to the server unless you configure a helper address for the server.
- **Gateway address** – The PowerConnect places the IP address of the interface that received the BootP or DHCP request in the request packet's Gateway Address field (sometimes called the Router ID field). When the server responds to the request, the server sends the response as a unicast packet to the IP address in the Gateway Address field. (If the client and server are directly attached, the Gateway ID field is empty and the server replies to the client using a unicast or broadcast packet, depending on the server.)

  By default, the PowerConnect uses the lowest-numbered IP address on the interface that receives the request as the Gateway address. You can override the default by specifying the IP address you want the PowerConnect to use.
- **Hop Count** – Each router that forwards a BootP or DHCP packet increments the hop count by 1. Routers also discard a forwarded BootP or DHCP request instead of forwarding the request if the hop count is greater than the maximum number of BootP or DHCP hops allows by the router. By default, the PowerConnect forwards a BootP or DHCP request if its hop count is four or less, but discards the request if the hop count is greater than four. You can change the maximum number of hops the PowerConnect will allow to a value from 1 – 15.

  NOTE
  The BootP or DHCP hop count is not the TTL parameter.

Configuring an IP helper address

The procedure for configuring a helper address for BootP or DHCP requests is the same as the procedure for configuring a helper address for other types of UDP broadcasts. Refer to "Configuring an IP helper address" on page 742.

Changing the IP address used for stamping BootP or DHCP requests

When the PowerConnect forwards a BootP or DHCP request, the PowerConnect "stamps" the Gateway Address field. The default value the PowerConnect uses to stamp the packet is the lowest-numbered IP address configured on the interface that received the request.

The BootP or DHCP stamp address is an interface parameter. Change the parameter on the interface that is connected to the BootP or DHCP client.

To change the IP address used for stamping BootP or DHCP requests received on interface 1/1, enter commands such as the following.

```
NetIron(config)# int e 1/1
NetIron(config-if-e1000-1/1)# ip bootp-gateway 109.157.22.26
```

These commands change the CLI to the configuration level for port 1/1, then change the BootP or DHCP stamp address for requests received on port 1/1 to 192.157.22.26. The PowerConnect will place this IP address in the Gateway Address field of BootP or DHCP requests that the PowerConnect receives on port 1/1 and forwards to the BootP or DHCP server.

Syntax: [no] ip bootp-gateway <*ip-addr*>

Changing the maximum number of hops to a BootP relay server

Each BootP or DHCP request includes a field Hop Count field. The Hop Count field indicates how many routers the request has passed through. When the PowerConnect receives a BootP or DHCP request, the PowerConnect looks at the value in the Hop Count field:

- If the hop count value is equal to or less than the maximum hop count the PowerConnect allows, the PowerConnect increments the hop count by one and forwards the request.
- If the hop count is greater than the maximum hop count the PowerConnect allows, the PowerConnect discards the request.

NOTE
The BootP or DHCP hop count is not the TTL parameter.

To modify the maximum number of BootP or DHCP hops, enter the following command.

```
NetIron(config)# bootp-relay-max-hops 10
```

This command allows the PowerConnect to forward BootP or DHCP requests that have passed through up to ten previous hops before reaching the PowerConnect.

Syntax: [no] bootp-relay-max-hops <*1-15*>

Default: 4

Filtering Martian addresses

Martian addresses are obviously invalid host or network addresses. They commonly are sent by improperly configured systems on the network. Martian address filtering allows the system to automatically filter out those invalid addresses. When Martian address filtering is enabled, the BGP protocol applies the Martian address filters to all in-bound routes as received from all neighbors. Unlike BGP protocol, IGP protocols will rely on the RTM (routing table manager) to do the route filtering.

If no match is found, the route is accepted. This will be the case for almost all routes. If a match is found, the route is discarded (default action - deny), unless the action is set to permit. Martian address filtering is in addition to normal BGP in-bound route policies.

To enable Martian address filtering, enter the following command.

```
NetIron(config)#  ip martian filtering-on
```

Syntax: [no] ip martian [vrf <*name*> **] filtering-on**

The **vrf** <*name*> option applies martian filtering to a specified VRF.

NOTE
Martian address filtering is disabled by default.

When Martian address filtering is first enabled, the router will automatically load the following default Martian addresses:

* 0.0.0.0/8
* 10.0.0.0/8
* 127.0.0.0/8
* 172.16.0.0/12
* 192.168.0.0/16
* 224.0.0.0/4
* 240.0.0.0/4

Adding, deleting or modifying Martian addresses

As described previously, there are a set number of Martian addresses that are loaded by default when Martian addressing is enabled. You can add, subtract or modify addresses that are filtered by martian addressing. Although there is no limit of the number of martian address can be configured, it's expected the size of martian address list should be small, generally less than 100. If the user adds a new martian address after routes are already learnt, they will be taken out of the routing table. Likewise if the user removes a martian address after routes are deleted from the routing table, they should be put back into the routing table.

To add an address to the Martian filtering list, use a command such as the following.

```
NetIron(config)# ip martian 192.168.0.0/16
```

Syntax: [no] ip martian [vrf <*name*> **]** <*destination-prefix/prefix-length*> **[permit]**

The <*destination-prefix/prefix-length*> variable specifies the address and the prefix range to apply the martian filtering to. The matching rule is for prefix range match. It includes exact match, or with a longer prefix length match. For example, if the Martian address rule is 192.168.0.0/16, then routes 192.168.0.0/16, and 192.168.1.0/24 are matches. However route 192.0.0.0/8 is not a match.

The **vrf** <*name*> option applies the modification to the martian filtering list to a specified VRF.

The **[no]** option removes an address from the martian filtering list.

The **[permit]** option changes the default action of a martian address filter to permit. In this case, a route matches the "permit" martian address is accepted by the routing table manager. This option is only used if a user wants to allow a prefix "hole" in an otherwise denied martian address.

The default Martian addresses are described in: "Filtering Martian addresses" on page 744

Examples

To remove a user defined Martian address or a system default Martian address, use the "no" form of the command.

```
NetIron(config)# no ip martian 0.0.0.0/8
```

The following example configuration, creates a “hole” for 192.168.1.0/24 in the martian address 192.168.0.0/16.

```
NetIron(config)# ip martian 192.168.1.0/24 permit
NetIron(config)# ip martian 192.168.0.0/16
```

To display the currently configured Martian addresses refer to “Displaying martian addressing information” on page 774.

IPv6 Over IPv4 tunnels in hardware

To enable communication between the isolated IPv6 domains using the IPv4 infrastructure, you can configure IPv6 over IPv4 tunnels.

Dell supports the following IPv6 over IPv4 tunneling in hardware mechanisms:

- Manually configured tunnels
- Automatic 6to4 tunnels

In general, a manually configured tunnel establishes a permanent link between routers in IPv6 domains, while the automatic tunnels establish a transient link that is created and taken down on an as-needed basis. (Although the feature name and description may imply otherwise, some configuration is necessary to set up an automatic tunnel.) Also, a manually configured tunnel has explicitly configured IPv4 addresses for the tunnel source and destination, while the automatic tunnels have an explicitly configured IPv4 address for the tunnel source and an automatically generated address for the tunnel destination.

These tunneling mechanisms require that the router at each end of the tunnel run both IPv4 and IPv6 protocol stacks. The routers running both protocol stacks, or dual-stack routers, can interoperate directly with both IPv4 and IPv6 end systems and routers.

The following features are not supported for IPv6 tunnel configuration at this time:

- Keep-alive
- Hitless upgrade
- Tunnels over MPLS or GRE

Configuring a IPv6 IP tunnel

To configure a IPv6 IP Tunnel, configure the following parameters:

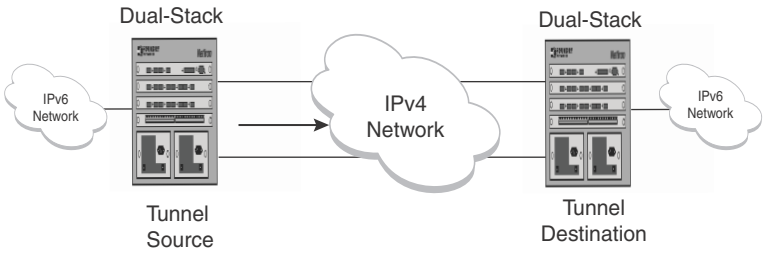
- CAM Restrictions
- Maximum Number of Tunnels (optional)
- Tunnel Interface
- Source Address or Source Interface for the Tunnel
- Destination address for the Tunnel

- IPv6 Encapsulation
- IP address for the Tunnel
- TTL Value (optional)
- TOS Value (optional)
- MTU Value (optional)

Configuring a manual IPv6 tunnel

You can use a manually configured tunnel to connect two isolated IPv6 domains. You should deploy this point-to-point tunnel mechanism if you need a permanent and stable connection.

FIGURE 131 Manually configured tunnel



Configuration notes on manual tunnels:

- The tunnel mode should be **ipv6ip** indicating that this is ipv6 manual tunnel.
- Both source and destination addresses needs to be configured on the tunnel.
- On the remote side you need to have exactly opposite source/destination pair.
- The tunnel destination should be reachable through the IPv4 backbone.
- The ipv6 address on the tunnel needs to be configured for the tunnel to come up.
- The tunnel source can be an IP address or interface name.
- Manual tunnels provide static point-point connectivity.
- Static routing on top of the tunnel is supported.
- IPv6 routing protocols including OSPFv3 and RIPng on top of the tunnel are supported.

NOTE
IPv6 IS-IS is not supported on top of the tunnel.

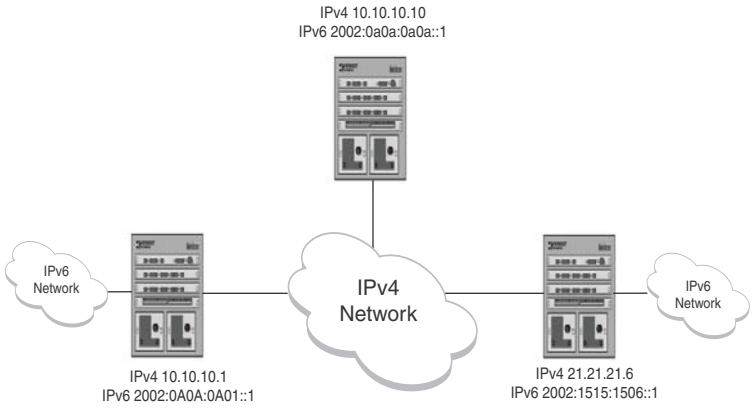
Configuring an automatic 6to4 tunnel

An automatic 6to4 tunnel establishes a transient link between IPv6 domains, which are connected by an IPv4 backbone. When needed, a device on which an automatic 6to4 tunnel is configured in one domain can establish a tunnel with another similarly configured device in another domain. When no longer needed, the devices take down the tunnel.

Instead of a manually configured tunnel destination, an automatic 6to4 tunnel constructs a globally unique 6to4 prefix, which determines the tunnel destination. The 6to4 prefix has the following format:

2002:<*ipv4-address*>::/48

When two domains need to communicate, a device creates a tunnel using the 6to4 prefix. The software automatically generates the 6to4 prefix by concatenating a configured static IPv6 prefix of 2002 with the destination device's globally unique IPv4 address. (Each device in an IPv6 domain that needs to communicate over an automatic 6to4 tunnel must have one globally unique IPv4 address, from which the globally unique 6to4 prefix is constructed.) After the communication ends, the tunnel is taken down.



Configuration notes on 6to4tunnels:

- This tunnel treats the IPv4 infrastructure as a virtual non-broadcast link and support multipoint connectivity.
- Tunnel mode must be configured as **ipv6ip 6to4.**
- Tunnel source must be configured.
- Tunnel destination is not configured on 6to4 tunnel explicitly, as the destination is specified as part of static nexthop or BGP nexthop.
- Static route with **2002::/16** MUST be configured.
- IPv6 address with **2002:A.B.C.D::/48** must be configured for the tunnel to come up (A.B.C.D is the tunnel source IP address).
- You can have 6to4 tunnel with multiple nexthops depending on the IPv6 nexthop used to forward the packets.
- With 6to4 tunnels, you can only use routing protocols (i.e. BGP+) that specify the nexthop in the configuration.
- OSPFv3, IPv6 IS-IS and RIPng are not supported on the 6to4 tunnels.
- Static routes can be used with 6to4 tunnels. If you use a static route to configure the nexthop, you MUST enable nexthop recursion in the system (ipv6 route next-hop-recursion).
- The 6to4 tunnel tries to resolve all the nexthops and programs the cam and pram entries needed. The IPv4 address in the nexthop should be reachable through the IPv4 network.

Example

In the below configuration:

- **10.10.10.1** is the tunnel source IP address
- **10.10.10.10** is the static nexthop
- **21.21.21.6** is I-BGP nexthop
- **22.22.22.6** is E-BGP nexthop

Static route Nexthop example:

- Create a static route pointing to the tunnel.

```
NetIron(config) #ipv6 route 2002::/16 tunnel 2 // Mandatory for 6to4 Configuration
NetIron(config) #ipv6 route next-hop-recursion // Mandatory with static nexthop
NetIron(config)# ipv6 route 3001::/64 2002:0a0a:0a0a::1 // Static Nexthop:
10.10.10.10
```

- Create a Source Interface - The remote node needs to have a similar route pointing to this node.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1 /1)ip address 10.10.10.1 255.255.255.0
```

- Create a 6to4 Tunnel configuration.

```
NetIron(config) interface tunnel 2
NetIron(config-tnif-2) tunnel mode ipv6ip 6to4
NetIron(config-tnif-1) tunnel source 10.10.10.1
NetIron(config-tnif-1) ipv6 address 2002:0a0a:0a01::1/64
```

Example : I-BGP Nexthop.

```
NetIron(config) router bgp
NetIron(config-bgp) local-as 100
NetIron(config-bgp) neighbor 2002:1515:1506::1 remote-as 100 // BGP Nexthop:
21.21.21.6
NetIron(config-bgp)# address-family ipv4 unicast
NetIron(config-bgp)# no neighbor 2002:1515:1506::1 activate
NetIron(config-bgp)# exit-address-family
NetIron(config-bgp)# address-family ipv4 multicast
NetIron(config-bgp)# exit-address-family
NetIron(config-bgp)# address-family ipv6 unicast
NetIron(config-bgp)# neighbor 2002:1515:1506::1 activate
NetIron(config-bgp)# exit-address-family
```

Example : E-BGP Nexthop.

```
NetIron(config)# router bgp
NetIron(config-bgp)# local-as 100
NetIron(config-bgp)# neighbor 2002:1616:1606::1 remote-as 101 // BGP Nexthop:
22.22.22.6
NetIron(config-bgp)# neighbor 2002:1616:1606::1 ebgp-multihop
NetIron(config-bgp)# address-family ipv4 unicast
NetIron(config-bgp)# no neighbor 2002:1515:1506::1 activate
NetIron(config-bgp)# exit-address-family
NetIron(config-bgp)# address-family ipv4 multicast
NetIron(config-bgp)# exit-address-family
NetIron(config-bgp)# address-family ipv6 unicast
NetIron(config-bgp)# neighbor 2002:1616:1606::1 activate
NetIron(config-bgp)# exit-address-family
```

Configuring the maximum number of tunnels supported

You can configure the routers to support a specified number of tunnels using the following command.

```
NetIron(config)# system-max ip-tunnels 512
NetIron(config)# write memory
```

Syntax: [no] system-max ip-tunnels <*number*>

The <*number*> variable specifies the number of IPv6 tunnels that can be supported on the PowerConnect router. The permissible range is 1 - 512. The default value is 256.

NOTE
You must write this command to memory and perform a system reload for this command to take effect.

Configuring a tunnel interface

To configure a tunnel interface, use a the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)
```

Syntax: [no] interface tunnel <*tunnel id*>

The <*tunnel-id*> variable is numerical value that identifies the tunnel being configured. Possible range is from 1 to the maximum configured tunnels in the system.

Configuring a source address or source interface for a tunnel interface

To configure a source address for a specific tunnel interface, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel source 35.0.8.108
```

To configure a source interface for a specific tunnel interface, enter the following command.

```
NetIron(config)# interface tunnel 100
NetIron(config-tnif-100)tunnel source ethernet 3/1
```

Syntax: [no] tunnel source <*ip-address*> **|** <*port-no*>

You can specify either of the following:

The <*ip-address*> variable is the source IP address being configured for the specified tunnel.
The <*port-no*> variable is the source slot/port of the interface being configured for the specified tunnel. When you configure a source interface, there must be at least one IP address configured on that interface. Otherwise, the interface will not be added to the tunnel configuration and an error message like the following will be displayed: " Error - Tunnel source interface 3/1 has no configured ip address.

It can be a physical or virtual interface (ve).

Configuring a destination address for a tunnel interface

To configure a destination address for a specific tunnel interface, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel destination 131.108.5.2
```

Syntax: [no] tunnel destination <*ip-address*>

The <*ip-address*> variable is destination IP address being configured for the specified tunnel.

Configuring a tunnel interface for IPv6 encapsulation

To configure a specified tunnel interface for IPv6 encapsulation, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel mode ipv6ip
```

Syntax: [no] tunnel mode ipv6ip < 6to4 | auto-tunnel >

The **6to4** parameter specifies automatic tunneling using 6 to 4.

The **auto-tunnel** parameter specifies automatic tunnel using ipv4 compatible ipv6 address.

Configuring an IP address for a tunnel interface

To configure an IP address for a specified tunnel interface, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)ipv6 address 1001:0a0a:0a01::1/64
```

Syntax: [no] ipv6 address <*ipv6-address*>

The <*ipv6-address*> variable is the IPv6 address being configured for the specified tunnel interface.

Configuring a TTL value

This is an optional parameter that allows you to set the Time-to-Live value for the outer IP header of the IPv6 tunnel packets.

To configure the TTL value, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel ttl 100
```

Syntax: [no] tunnel ttl <*ttl-value*>

The <*ttl-value*> variable specifies a TTL value for the outer IP header. Possible values are 1 - 255. The default value is 255.

Configuring a TOS value

This is an optional parameter that allows you to set the TOS value for the outer IP header of the GRE tunnel packets.

To configure the TOS value, enter the following command.

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel tos 100
```

Syntax: [no] tunnel ttl <*tos-value*>

The <*tos-value*> variable specifies a TOS value for the outer IP header. Possible values are 1 - 255. The default value is 0.

Configuring IPv6 session enforce check

You can enable the IPv6 session enforce check by using the **ipv6-session-enforce-check** command. When an IPv6 packet arrives and this feature is enabled, the system tries to match the Ipv6 packet source and destination address pair with the tunnel configured destination and source pair. If the pairs do not match, the packet is dropped in hardware.

To configure the IPv6 session enforce check,go to the IP tunnel policy context and enter the **ipv6-session-enforce-check** command.

```
NetIron@MLXe1(config)#ip-tunnel-policy
NetIron@MLXe1(config-ip-tunnel-policy)#ipv6-session-enforce-check
```

Syntax: [no] ipv6-session-enforce-check

To disable the IPv6 session enforce check, use the **no** form of this command. This command is disabled by default. You might have to write the configuration to memory and reload the system when the configuration of this command is changed because a one-time creation of a source-ingress CAM partition is necessary. The system prompts you if the memory write and reload are required.

The first-time execution of certain commands necessitates the creation of a source-ingress CAM partition, after which you write to memory and reload. These commands are **gre-session-enforce-check**, **ipv6-session-enforce-check**, and **accounting-enable**. After this CAM partition is created, it is not necessary to follow either of the other two commands with a memory write and reload.

NOTE
The ipv6-sessions-enforce-check is not supported for 6to4 automatic tunnels.

Configuring a maximum MTU value for a tunnel interface

This command allows you to set an MTU value for packets entering the tunnel. Packets that exceed either the default MTU value of 1480 bytes or the value that you set using this command are sent back to the source.

The following command allows you to change the MTU value for packets transiting "tunnel 1".

```
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel mtu 1500
```

Syntax: [no] tunnel mtu <*packet-size*>

The <*packet-size*> variable specifies the maximum MTU size in bytes for the packets transiting the tunnel.

NOTE
To prevent packet loss after the 20 byte IP header is added, make sure that any physical interface that is carrying IPv6 tunnel traffic has an IP MTU setting at least 24 bytes greater than the tunnel MTU setting.

Displaying IPv6 tunneling information

You can display IPv6 Tunneling Information using the **show ip-tunnels**, **show ipv6 interface**, **show ipv6 route** and **show interface tunnel** commands as shown in the following:

Displaying tunnel information

For example, to tunnel information for tunnel 2, enter the following command at any level of the CLI.

```
NetIron# show ip-tunnels 2
IPv6 tnnl 2 UP   : src_ip 192.211.2.1, dst_ip 192.212.2.1
   TTL 255, TOS 0, NHT 0, MTU 1480
```

Syntax: show ip tunnels <*number*>

The <*number*> parameter indicates the tunnel interface number for which you want to display information.

This display shows the following information.

TABLE 114 Show IP tunnel display information

This field...	Displays...
Ipv6 tnnl <*UP* | *DOWN*>	The status of the tunnel interface can be one of the following: • up – The tunnel interface is functioning properly. • down – The tunnel interface is not functioning and is down.
src_ip	The tunnel source can an IPv4 address.
dst_ip	The tunnel destination can an IPv4 address.
TTL	The TTL value configured for the outer IP header. Possible values are 1 - 255.
TOS	The TOS value configured for the outer IP header. Possible values are 1 - 255.
NHT	The nextHop Table index value.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying tunnel interface information

For example, to display status and configuration information for tunnel interface 1, enter the following command at any level of the CLI.

```
NetIron# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source ethernet 3/5
  Tunnel destination is not configured
  Tunnel mode ipv6ip auto-tunnel
  No port name
  MTU 1500 bytes
```

Syntax: show interfaces tunnel <*number*>

The <*number*> parameter indicates the tunnel interface number for which you want to display information.

This display shows the following information.

TABLE 115 IPv6 tunnel interface information

This field...	Displays...
Tunnel interface status	The status of the tunnel interface can be one of the following: • up – The tunnel interface is functioning properly. • down – The tunnel interface is not functioning and is down.
Line protocol status	The status of the line protocol can be one of the following: • up – The line protocol is functioning properly. • down – The line protocol is not functioning and is down.
Hardware is tunnel	The interface is a tunnel interface.
Tunnel source	The tunnel source can be one of the following: • An IPv4 address • The IPv4 address associated with an interface or port.
Tunnel destination	The tunnel destination can an IPv4 address.

TABLE 115 IPv6 tunnel interface information (Continued)

This field...	Displays...
Tunnel mode	The tunnel mode can be one the following: • ipv6ip auto-tunnel – Indicates an automatic IPv4-compatible tunnel. • ipv6ip 6to4 – Indicates an automatic 6to4 tunnel.
Port name	The port name configured for the tunnel interface.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying interface level IPv6 settings

To display Interface level IPv6 settings for tunnel interface 1, enter the following command at any level of the CLI.

```
NetIron# show ipv6 inter tunnel 1
Interface Tunnel 1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::3:4:2 [Preferred]
  Global unicast address(es):
    1001::1 [Preferred],  subnet is 1001::/64
    1011::1 [Preferred],  subnet is 1011::/64
  Joined group address(es):
    ff02::1:ff04:2
    ff02::5
    ff02::1:ff00:1
    ff02::2
    ff02::1
  MTU is 1480 bytes
  ICMP redirects are enabled
  No Inbound Access List Set
  No Outbound Access List Set
  OSPF enabled
```

The display command above reflects the following configuration.

```
NetIron# show running-config interface tunnel 1
!
interface tunnel 1
 port-name ManualTunnel1
 tunnel mode ipv6ip
 tunnel source loopback 1
 tunnel destination 2.1.1.1
 ipv6 address fe80::3:4:2 link-local
 ipv6 address 1011::1/64
 ipv6 address 1001::1/64
 ipv6 ospf area 0
```

Displaying IP information

You can display the following IP configuration information statistics:

- **Global IP parameter settings** – refer to "Displaying global IP configuration information" on page 755.
- **IP interfaces** – refer to "Displaying IP interface information" on page 756.
- **ARP entries** – refer to "Displaying ARP entries" on page 759.

- **Static ARP entries** – refer to "Displaying ARP entries" on page 759.
- **IP forwarding cache** – refer to "Displaying the forwarding cache" on page 761.
- **IP route table** – refer to "Displaying the IP route table" on page 763.
- **IP traffic statistics** – refer to "Displaying IP traffic statistics" on page 768.

The sections below describe how to display this information.

In addition to the information described below, you can display the following IP information:

- **RIP information** – refer to "Displaying RIP Information" on page 852.
- **OSPF information** – refer to "Displaying OSPF database information" on page 921.
- **BGP4 information** – refer to "Displaying BGP4 information" on page 1104.
- **PIM information** – refer to "Displaying PIM Sparse configuration information and statistics" on page 1174.

Displaying global IP configuration information

To display IP configuration information, enter the following command at any CLI level.

```
NetIron> show ip

Global Settings
  IP CAM Mode: dynamic IPVPN CAM Mode: static
  ttl: 64, arp-age: 10, bootp-relay-max-hops: 4, icmp-error-rate: 400
  IP Router-Id: 5.5.5.5
  enabled : UDP-Broadcast-Forwarding ICMP-Redirect Source-Route Load-Sharing
  RARP BGP4 OSPF
  disabled: Directed-Broadcast-Forwarding drop-arp-pending-packets IRDP Proxy
  -ARP RPF-Check RPF-Exclude-Default RIP IS-IS VRRP VRRP-Extended VSRP
Configured Static Routes: 31
Configured Static Mroutes: 30
```

Syntax: show ip

NOTE
This command has additional options, which are explained in other sections in this guide, including the sections below this one.

This display shows the following information.

TABLE 116 CLI display of global IP configuration information

This field...	Displays...
Global settings	
ttl	The Time-To-Live (TTL) for IP packets. The TTL specifies the maximum number of router hops a packet can travel before reaching the PowerConnect. If the packet's TTL value is higher than the value specified in this field, the router drops the packet. To change the maximum TTL, refer to "Changing the TTL threshold" on page 709.
arp-age	The ARP aging period. This parameter specifies how many minutes an inactive ARP entry remains in the ARP cache before the router ages out the entry. To change the ARP aging period, refer to "Changing the ARP aging period" on page 688.

TABLE 116 CLI display of global IP configuration information (Continued)

This field...	Displays...
bootp-relay-max-hops	The maximum number of hops away a BootP server can be located from the router and still be used by the router's clients for network booting. To change this value, refer to "Changing the maximum number of hops to a BootP relay server" on page 744.
router-id	The 32-bit number that uniquely identifies the router. By default, the router ID is the numerically lowest IP interface configured on the router. To change the router ID, refer to "Changing the router ID" on page 684.
enabled	The IP-related protocols that are enabled on the router.
disabled	The IP-related protocols that are disabled on the router.

Displaying IP interface information

To display IP interface information, enter the following command at any CLI level.

```
NetIron(config)# show ip interface

Interface       IP-Address      OK?  Method    Status                Protocol
Ethernet 1/1    207.95.6.173    YES  NVRAM     up                    up
Ethernet 1/2    3.3.3.3         YES  manual    up                    up
Loopback 1      1.2.3.4         YES  NVRAM     down                  down
```

Syntax: show ip interface [ethernet <*slot/port*>**] | [loopback** <*num*>**] | [ve** <*num*>**]**

This display shows the following information.

TABLE 117 CLI display of interface IP configuration information

This field...	Displays...
Interface	The type and the slot and port number of the interface.
IP-Address	The IP address of the interface. **NOTE:** If an "s" is listed following the address, this is a secondary address. When the address was configured, the interface already had an IP address in the same subnet, so the software required the "secondary" option before the software could add the interface.
OK?	Whether the IP address has been configured on the interface.
Method	Whether the IP address has been saved in NVRAM. If you have set the IP address for the interface in the CLI, but have not saved the configuration, the entry for the interface in the Method field is "manual".
Status	The link status of the interface. If you have disabled the interface with the **disable** command, the entry in the Status field will be "administratively down". Otherwise, the entry in the Status field will be either "up" or "down".
Protocol	Whether the interface can provide two-way communication. If the IP address is configured, and the link status of the interface is up, the entry in the protocol field will be "up". Otherwise the entry in the protocol field will be "down".

Displaying IP interface information for a specified interface

To display detailed IP information for a specific interface, enter a command such as the following.

```
NetIron# show ip interface ethernet e 3/1
Interface Ethernet 3/1 (80)
  port enabled
  port state: UP
  ip address: 13.1.1.2/24
  Port belongs to VRF: default
  encapsulation: ETHERNET, mtu: 1500
  MAC Address 0004.80a0.4050
  directed-broadcast-forwarding: disabled
  No inbound ip access-list is set
  No outbound ip access-list is set
  No Helper Addresses are configured.
  RPF mode: None  RPF Log: Disabled
  0 unicast RPF drop    0 unicast RPF suppressed drop
  RxPkts:  1200  TxPkts:  1200
  RxBytes: 60000 TxBytes: 60000
```

The PowerConnect software supports IPv4 and IPv6 packet and byte counters. The contents of these counters is displayed for a defined port as the result of the show ip interface ethernet command. In the above example, the fields in bold text display this content. Table 118 describes each of the fields that display interface counter statistics.

TABLE 118 Interface counter display statistics

This field...	Displays...
Interface	The interface that counter statistics are being displayed for.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying interface counters for all ports

The PowerConnect supports IPv4 and IPv6 packet and byte counters. The contents of these counters can be displayed for all ports on a router or per-port. Output from the **show ip interface ethernet** command has been enhanced to include packet and byte counter information on a per-port basis. This is described in "Displaying interface counters for all ports" on page 757.

Commands have been added under IPv4 and IPv6 to display the interface counters for all ports on a router. The following example uses the **show ip interface counters** command to display to packet and byte counter information for all ports.

```
NetIron# show ip interface counters
Interface      RxPkts     TxPkts      RxBytes    TxBytes
eth 3/1         1200      1200        600000     60000
eth 3/2         500       500         25000      25000
```

Syntax: show ip interface counters

Default byte counters include the 20-byte per-packet Ethernet overhead. You can configure an PowerConnect router to exclude the 20-byte per-packet Ethernet overhead from byte accounting by configuring the **vlan-counter exclude-overhead** command.Table 118 describes each of the fields that display interface counter statistics.

TABLE 119 Interface counter display statistics

This field...	Displays...
Interface	The interface that counter statistics are being displayed for.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Clearing the interface counters

Use the following command to clear all interface counters on a router.

```
NetIron# clear ip interface counters
```

Syntax: clear ip interface counters

Use the following command to clear the interface counters for a specified port.

```
NetIron# clear ip interface ethernet 3/2
```

Syntax: clear ip interface ethernet <*port-number*>

The **port-number** variable specifies the slot and port number that you want to clear the interface counters for.

Displaying interface name in Syslog

By default an interface's slot number (if applicable) and port number are displayed when you display Syslog messages. You can display the name of the interface instead of its number by entering a command such as the following.

```
NetIron(config)# ip show-portname
```

This command is applied globally to all interfaces on the PowerConnect.

Syntax: [no] Ip show-portname

When you display the messages in the Syslog, you see the interface name under the Dynamic Log Buffer section. The actual interface number is appended to the interface name. For example, if the interface name is "lab" and its port number is "2", you see "lab2" displayed as in the example below.

```
NetIron># show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
         I=informational N=notification W=warning
Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet Lab2, state up
Dec 15 18:45:15:I:Warm start
```

Displaying ARP entries

You can display the ARP cache and the static ARP table. The ARP cache contains entries for devices attached to the PowerConnect. The static ARP table contains the user-configured ARP entries. An entry in the static ARP table enters the ARP cache when the entry's interface comes up.

The tables require separate display commands.

Displaying the ARP cache

To display the contents of the ARP cache, enter the following command at any CLI level.

```
NetIron# show arp

Total number of ARP entries: 5
     IP Address          MAC Address         Type        Age     Port
1     207.95.6.102        0800.5afc.ea21      Dynamic     0         6
2     207.95.6.18         00a0.24d2.04ed      Dynamic     3         6
3     207.95.6.54         00a0.24ab.cd2b      Dynamic     0         6
4     207.95.6.101        0800.207c.a7fa      Dynamic     0         6
5     207.95.6.211        00c0.2638.ac9c      Dynamic     0         6
6     30.30.30.15         none                Pending     0         v1
```

Syntax: **show arp** [**ethernet** <*slot/port*> | **mac-address** <*xxxx.xxxx.xxxx*> [<*mask*>] | <*ip-addr*> [<*ip-mask*>]] [<*num*>] [| **begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

The **ethernet** <*slot*>/<*portnum*> parameter lets you restrict the display to entries for a specific port.

The **mac-address** <*xxxx.xxxx.xxxx*> parameter lets you restrict the display to entries for a specific MAC address.

The <*mask*> parameter lets you specify a mask for the **mac-address** <*xxxx.xxxx.xxxx*> parameter to display entries for multiple MAC addresses. Specify the MAC address mask as fs and 0s, where fs are significant bits.

The <*ip-addr*> and <*ip-mask*> parameters let you restrict the display to entries for a specific IP address and network mask. Specify the IP address masks in standard decimal mask format (for example, 255.255.0.0).

NOTE
The <*ip-mask*> parameter and <*mask*> parameter perform different operations. The <*ip-mask*> parameter specifies the network mask for a specific IP address, whereas the <*mask*> parameter provides a filter for displaying multiple MAC addresses that have specific values in common.

The <*num*> parameter lets you display the table beginning with a specific entry number.

NOTE
The entry numbers in the ARP cache are not related to the entry numbers for static ARP table entries.

This display shows the following information. The number in the left column of the CLI display is the row number of the entry in the ARP cache. This number is not related to the number you assign to static MAC address entries in the static ARP table.

TABLE 120 CLI display of ARP cache

This field...	Displays...
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Type	The type, which can be one of the following: • Dynamic – The PowerConnect router learned the entry from an incoming packet. • Static – The PowerConnect router loaded the entry from the static ARP table when the device for the entry was connected to the PowerConnect. • Pending – The PowerConnect router added the entry to the ARP table and is in the process of sending a series of ARP requests to determine if it is a valid entry.
Age	The number of minutes the entry has remained unused. If this value reaches the ARP aging period, the entry is removed from the table. To display the ARP aging period, refer to "Displaying global IP configuration information" on page 755. To change the ARP aging interval, refer to "Changing the ARP aging period" on page 688. **NOTE:** Static entries do not age out.
Port	The port on which the entry was learned.

Displaying the static ARP table

To display the static ARP table, enter the following command at any CLI level.

```
NetIron# show ip static-arp

Total no. of entries: 4
  Index   IP Address          MAC Address         Port      VLAN  ESI
  1       1.1.1.1             0001.0001.0001      1/1
  2       6.6.6.2             0002.0002.0002      1/2
```

Syntax: **show ip static-arp [ethernet** <*slot*>/<*portnum*> **| mac-address** <*xxxx.xxxx.xxxx*> [<*mask*>] **|** <*ip-addr*> [<*ip-mask*>]] [<*num*>] [**| begin** <*expression*> **| exclude** <*expression*> **| include** <*expression*>]

For information on the command syntax, see the syntax of the **show arp** command under "Displaying the ARP cache" on page 759.

TABLE 121 CLI display of static ARP table

This field...	Displays...
Index	The number of this entry in the table.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.

TABLE 121 CLI display of static ARP table (Continued)

This field...	Displays...
Port	The port attached to the device the entry is for. In the case of a multi-port static ARP, this will display a single port followed by an ellipsis, and the full list of ports will be displayed on the line below.
VLAN	VLAN associated with this entry, if any.
ESI	Ethernet Service Instance (ESI) associated with this entry, if any.

Displaying the forwarding cache

To display the IP Forwarding Cache for directly connected hosts, enter the following command.

```
NetIron> show ip cache
Cache Entry Usage on LPs:
Module    Host   Network      Free       Total
15         6      6           204788     204800
```

Syntax: **show ip cache** [<*ip-addr*>] [**| begin** <*expression*> **| exclude** <*expression*> **| include** <*expression*>]

The <*ip-addr*> parameter displays the cache entry for the specified IP address.

The **show ip cache** command shows the forwarding cache usage on each interface module CPU. The CPU on each interface module builds its own forwarding cache, depending on the traffic. To see the forwarding cache of a particular interface module, use the **rconsole**.

```
NetIron>rconsole 15
Connecting to slave CPU 15/1... (Press CTRL-Shift-6 X to exit)
rconsole-15/1@LP>show ip cache
Total number of host cache entries 3
D: Dynamic P:Permanent, F:Forward U:Us C:Conected Network
W:Wait ARP  I:ICMP Deny  K:Drop R:Frament S:Snap Encap  N:CAMInvalid
    IP Address    Next Hop     MAC              Type    Port    VLAN     Pri
1   30.1.0.0      DIRECT       0000.0000.0000    PU      2/5     n/a      0
2   20.1.0.0      DIRECT       0125.0a57.1c02    D       3/5     n/a      0
3   7.7.7.3       DIRECT       0000.0000.0000    PU      4/2     12       1
```

You also use the **rconsole** to display the IP Forwarding Cache for network entries.

```
NetIron>rconsole 15
Connecting to slave CPU 15/1... (Press CTRL-Shift-6 X to exit)
rconsole-15/1@LP>show ip network
Total number of host cache entries 3
D: Dynamic P:Permanent, F:Forward U:Us C:Conected Network
W:Wait ARP  I:ICMP Deny  K:Drop R:Frament S:Snap Encap  N:CAMInvalid
    IP Address       Next Hop     MAC              Type    Port    VLAN     Pri
1   0.0.0.0/0        DIRECT       0000.0000.0000    PK              n/a      0
2   20.1.1.0/24      DIRECT       0000.0000.0000    PC              n/a      0
3   40.40.40.0/24    30.1.1.10    0000.0000.0033    PF      15/14   154      1
```

The **show ip cache** and **show ip network** commands entered on the rconsole display the following information.

TABLE 122 CLI display of IP forwarding cache

This field...	Displays...
IP Address	The IP address of the destination.
Next Hop	The IP address of the next-hop router to the destination. This field contains either an IP address or the value DIRECT. DIRECT means the destination is either directly attached or the destination is an address on this device. For example, the next hop for loopback addresses and broadcast addresses is shown as DIRECT.
MAC	The MAC address of the destination. **NOTE:** If the entry is type U (indicating that the destination is this device), the address consists of zeroes.
Type	The type of host entry, which can be one or more of the following: • D – Dynamic • P – Permanent • F – Forward • U – Us • C – Complex Filter • W – Wait ARP • I – ICMP Deny • K – Drop • R – Fragment • S – Snap Encap
Port	The port through which this device reaches the destination. For destinations that are located on this device, the port number is shown as “n/a”.
VLAN	Indicates the VLANs the listed port is in.
Pri	The QoS priority of the port or VLAN.

Dual Active Console

The Dual Active Console command enables the standby terminal console to mirror the features of the active console, such that the standby console appears as active console itself. Hence, you can manage the system from either active or standby console and it will not be necessary to switch the console cable after the active-standby management module switchover.

To enable this feature, enter the following command,

```
NetIron(config)#dual-act
NetIron(config)#wr mem
Write startup-config done.
NetIron(config)#
```

To disable this feature, enter the following command,

```
NetIron(config)#no dual-act
NetIron(config)#wr mem
Write startup-config done.
NetIron(config)#
```

Displaying the IP route table

To display the IP route table, enter the **show ip route** command at any CLI level.

```
NetIron# show ip route
Total number of IP routes: 4
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
        Destination        Gateway         Port        Cost       Type Uptime
1       10.0.0.0/24        DIRECT          eth 1/1     0/0        D    45m18s
2       11.0.0.0/24        DIRECT          eth 1/2     0/0        D    1h0m
3       90.0.0.0/24        10.0.0.2        eth 1/1     1/1        S    13m18s
4       100.0.0.0/24       10.0.0.2        eth 1/1     1/1        S    2m42s
```

Syntax: **show ip route** <*num*> | [<*ip-addr*> [<*ip-mask*>] **[debug | detail | longer]**] | **connected | bgp | isis | ospf | rip | static | [summary] | nexthop** [<*nexthop_id*> **[ref-routes]] | [begin** <*expression*> **| exclude** <*expression*> **| include** <*expression*>]

The <*num*> option display the route table entry whose row number corresponds to the number you specify. For example, if you want to display the tenth row in the table, enter "10".

The <*ip-addr*> parameter displays the route to the specified IP address.

The <*ip-mask*> parameter lets you specify a network mask or, if you prefer CIDR format, the number of bits in the network mask. If you use CIDR format, enter a forward slash immediately after the IP address, then enter the number of mask bits (for example: 209.157.22.0/24 for 209.157.22.0 255.255.255.0).

The **longer | detail | debug** parameter applies only when you specify an IP address and mask. This option displays only the routes for the specified IP address and mask.

The **bgp** option displays the BGP4 routes.

The **connected** option displays only the IP routes that are directly attached to the PowerConnect.

The **ospf** option displays the OSPF routes.

The **rip** option displays the RIP routes.

The **isis** option displays the RIP routes.

The **static** option displays only the static IP routes.

The **nexthop** option displays next-hop information for all next hops in the routing table or for a specific entry.

Showing route details by IP address

You can display detailed information about a route by providing the IP address and using the **detail** option, as the following example illustrates.

```
NetIron@MLXe1>show ip route 41.1.1.2 detail
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       41.1.1.0/24        DIRECT          eth 1/15    0/0           D    7h11m
        Nexthop Entry ID:14, Paths: 1, Ref_Count:1/1
1       41.1.1.0/24        41.1.1.2        eth 1/15    115/20        IL2  7h11m
        41.1.1.0/24        129.0.0.18      eth 4/11    115/20        IL2  7h11m
        41.1.1.0/24        129.0.0.30      eth 4/7     115/20        IL2  7h11m
        41.1.1.0/24        129.0.0.34      eth 4/14    115/20        IL2  7h11m
        Nexthop Entry ID:68343, Paths: 4, Ref_Count:8/21
D:Dynamic  P:Permanent  F:Forward  U:Us  C:Connected Network
W:Wait ARP  I:ICMP Deny  K:Drop  R:Fragment  S:Snap Encap N:CamInvalid
Module S1:
      IP Address         Next Hop        MAC              Type  Port  Vlan  Pri
      41.1.1.0/24         DIRECT          0000.0000.0000     PC    n/a          0
      OutgoingIf  ArpIndex PPCR_ID   CamLevel   Parent  DontAge Index
       eth 1/15     65535    1:2        1              0    69203192   38
      U_flags   Entry_flags  Age   Cam:Index           Trunk_fid      Ecmp_count
       0000e220                0     0x1a8fc (L3, right)   0x00000(     0) 0
        CAM Entry Flag: 00000003H
        PPCR : 1:2 CIDX: 0x1a8fc (L3, right) (IP_NETWORK: 0x68703)
         PPCR : 1:1 CIDX: 0x1a8fc (L3, right) (IP_NETWORK: 0x68703)
```

Syntax: show ip route <*ip_addr*> **detail**

The IP address can be just the IP address but can also include shorthand for the mask: ip-address/prefix-length.

Using the summary option

The **summary** option displays a summary of the information in the IP route table. After the **summary** keyword, the pipe symbol (|) points to three options for modifying the presentation of the summary information, as follows:

- **begin** lets you start the display with the first matching line.
- **exclude** lets you exclude matching lines from the display.
- **include** lets you include matching lines in the display.

The default routes are displayed first.

Using the connected option

Here is an example of how to use the **connected** option. To display only the IP routes that go to devices directly attached to the PowerConnect.

```
NetIron(config)# show ip route connected
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
      Destination       Gateway          Port   Cost   Type   Uptime
1     209.157.22.0/24     0.0.0.0           4/11    1       D      1h0m
```

Notice that the route displayed in this example has “D” in the Type field, indicating the route is for a directly connected device.

Using the static option

Here is an example of how to use the **static** option. To display only the static IP routes.

```
NetIron(config)# show ip route static
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
      Destination       Gateway           Port   Cost   Type  Uptime
1     192.144.33.11/32  209.157.22.12     1/1    2      S     1h0m
```

Notice that the route displayed in this example has “S” in the Type field, indicating the route is static.

Using the longer option

Here is an example of how to use the **longer** option. To display only the routes for a specified IP address and mask, enter a command such as the following.

```
NetIron(config)# show ip route 209.159.0.0/16 longer
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
   Destination       Gateway           Port   Cost   Type  Uptime
52 209.159.38.0/24   207.95.6.101      1/1    1      S     45m18s
53 209.159.39.0/24   207.95.6.101      1/1    1      S     1h0m
54 209.159.40.0/24   207.95.6.101      1/1    1      S     45m18s
55 209.159.41.0/24   207.95.6.101      1/1    1      S     1h0m
56 209.159.42.0/24   207.95.6.101      1/1    1      S     13m18s
```

This example shows all the routes for networks beginning with 209.159. The mask value and **longer** parameter specify the range of network addresses to be displayed. In this example, all routes within the range 209.159.0.0 – 209.159.255.255 are listed.

Using the summary option

The **summary** option displays a summary of the information in the IP route table. The following is an example of the output from this command.

```
NetIron# show ip route summary

IP Routing Table - 35 entries:
  6 connected, 28 static, 0 RIP, 1 OSPF, 0 BGP, 0 ISIS, 0 MPLS
  Number of prefixes:
  /0: 1 /16: 27 /22: 1 /24: 5 /32: 1
```

Syntax: show ip route summary

In this example, the IP route table contains 35 entries. Of these entries, 6 are directly connected devices, 28 are static routes, and 1 route was calculated through OSPF. One of the routes has a zero-bit mask (this is the default route), 27 have a 22-bit mask, 5 have a 24-bit mask, and 1 has a 32-bit mask.

Using the nexthop option

You can display next-hop information for all next hops in the routing table or for a specific entry. For the first example, use the **show ip route nexthop** command to display all the next-hop entries, and then use the option to display the next hop for a specific table entry.

```
NetIron#show ip route nexthop
Total number of IP nexthop entries: 30; Forwarding Use: 24
       NextHopIp        Port             RefCount    ID      Age
1      0.0.0.0         mgmt 1         0/1        1536    80682
2      0.0.0.0         eth 1/15       1/1        14      80632
3      0.0.0.0         eth 1/16       1/1        15      16626
4      0.0.0.0         eth 1/18       1/1        17      16626
5      0.0.0.0         eth 1/43       1/1        42      35923
6      0.0.0.0         eth 1/47       1/1        46      80641
7      0.0.0.0         eth 2/2        1/1        49      16630
8      0.0.0.0         eth 2/4        1/1        51      16630
9      41.1.1.2        eth 1/15       0/2        68347   16620
       41.1.2.2        eth 1/18
       129.0.0.18      eth 4/11
       129.0.0.25      eth 4/9
10     41.1.1.2        eth 1/15       0/3        68352   16615
       129.0.0.6       eth 4/4
       129.0.0.10      eth 2/2
       129.0.0.21      eth 4/1
11     0.0.0.0         eth 4/1        1/1        144     16624
12     0.0.0.0         eth 4/3        1/1        146     16641
13     0.0.0.0         eth 4/4        1/1        147     16624
14     0.0.0.0         eth 4/6        1/1        149     16624
15     0.0.0.0         eth 4/7        1/1        150     16641
```

Syntax: show ip route nexthop [nexthop_id]

The <*nexthop_id*> is under the column labeled ID in the output of the **show ip route nexthop** command. For example, use nexthop ID 1536 from the first row of the preceding example to show only that entry.

```
NetIron#show ip route nexthop 1536
       NextHopIp        Port             RefCount      ID      Age
1      0.0.0.0          mgmt 1           0/1           1536    80685
```

Displaying IP routes with nexthop ID

By using the **nexthop** option with the **ref-routes** keyword, you can display IP routes in the forwarding table that refer to the specified nexthop entry, as the following example illustrates (using nexthop ID 65575).

```
NetIron#show ip route nexthop 65537 ref-routes
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       40.1.1.1/32        15.1.1.1        eth 1/11    115/10        IL2  7h51m
2       50.1.1.0/24        15.1.1.1        eth 1/11    115/10        IL2  7h51m
3       100.1.1.1/32       15.1.1.1        eth 1/11    115/40        IL2  7h51m
```

Syntax: show ip route nexthop [<*nexthop_id*> **[ref-routes]]**

Description of command output fields

The following table lists the information in the **show ip route** output when you use no optional arguments.

TABLE 123 CLI display of IP route table

This field...	Displays...
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router.
Port	The port through which this router sends packets to reach the route's destination.
Cost	The route's cost.
Type	The route type, which can be one of the following: • B – The route was learned from BGP. • D – The destination is directly connected to this PowerConnect. • R – The route was learned from RIP. • S – The route is a static route. • * – The route is a candidate default route. • O – The route is an OSPF route. Unless you use the ospf option to display the route table, "O" is used for all OSPF routes. If you do use the ospf option, the following type codes are used: • O – OSPF intra area route (within the same area). • IA – The route is an OSPF inter area route (a route that passes from one area into another). • E1 – The route is an OSPF external type 1 route. • E2 – The route is an OSPF external type 2 route.
Uptime	The amount of time since the route was last modified. The format of this display parameter may change depending upon the age of the route to include the seconds (s), minutes (m), hours (h), and days (d), as described in the following: 400d – Only days (d) displayed 20d23h – days (d) and hours (h) displayed 14h33m – hours (h) and minutes (m) displayed 10m59s – minutes (m) and seconds (s) displayed

Clearing IP routes

If needed, you can clear the entire route table or specific individual routes.

To clear all routes from the IP route table.

```
NetIron# clear ip route
```

To clear route 209.157.22.0/24 from the IP routing table.

```
NetIron# clear ip route 209.157.22.0/24
```

Syntax: clear ip route [*<ip-addr> <ip-mask>* | *<ip-addr>/<mask-bits>*]

Displaying IP traffic statistics

To display IP traffic statistics, enter the following command at any CLI level.

NOTE
In the PowerConnect, only those packets that are forwarded or generated by the CPU are included in the IP traffic statistics. Hardware forwarded packets are not included.

```
NetIron#show ip traffic
IP Statistics
  139 received, 145 sent, 0 forwarded
  0 filtered, 0 fragmented, 0 reassembled, 0 bad header
  0 no route, 0 unknown proto, 0 no buffer, 0 other errors

ICMP Statistics
Received:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp reply, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation
Sent:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp reply, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation

UDP Statistics
  1 received, 0 sent, 1 no port, 0 input errors

TCP Statistics
  0 active opens, 0 passive opens, 0 failed attempts
  0 active resets, 0 passive resets, 0 input errors
  138 in segments, 141 out segments, 4 retransmission
```

Syntax: show ip traffic

The **show ip traffic** command displays the following information.

TABLE 124 CLI display of IP traffic statistics

This field...	Displays...
IP statistics	
received	The total number of IP packets received by the device.
sent	The total number of IP packets originated and sent by the device.
forwarded	The total number of IP packets received by the device and forwarded to other devices.
filtered	The total number of IP packets filtered by the device.
fragmented	The total number of IP packets fragmented by this device to accommodate the IP MTU of this device or of another device.
reassembled	The total number of fragmented IP packets that this device re-assembled.
bad header	The number of IP packets dropped by the device due to a bad packet header.
no route	The number of packets dropped by the device because there was no route.
unknown proto	The number of packets dropped by the device because the value in the Protocol field of the packet header is unrecognized by this device.

TABLE 124 CLI display of IP traffic statistics (Continued)

This field...	Displays...
no buffer	This information is used by Dell customer support.
other errors	The number of packets that this device dropped due to error types other than the types listed above.
ICMP statistics The ICMP statistics are derived from RFC 792, "Internet Control Message Protocol", RFC 950, "Internet Standard Subnetting Procedure", and RFC 1256, "ICMP Router Discovery Messages". Statistics are organized into Sent and Received. The field descriptions below apply to each.	
total	The total number of ICMP messages sent or received by the device.
errors	This information is used by Dell customer support.
unreachable	The number of Destination Unreachable messages sent or received by the device.
time exceed	The number of Time Exceeded messages sent or received by the device.
parameter	The number of Parameter Problem messages sent or received by the device.
source quench	The number of Source Quench messages sent or received by the device.
redirect	The number of Redirect messages sent or received by the device.
echo	The number of Echo messages sent or received by the device.
echo reply	The number of Echo Reply messages sent or received by the device.
timestamp	The number of Timestamp messages sent or received by the device.
timestamp reply	The number of Timestamp Reply messages sent or received by the device.
addr mask	The number of Address Mask Request messages sent or received by the device.
addr mask reply	The number of Address Mask Replies messages sent or received by the device.
irdp advertisement	The number of ICMP Router Discovery Protocol (IRDP) Advertisement messages sent or received by the device.
irdp solicitation	The number of IRDP Solicitation messages sent or received by the device.
UDP statistics	
received	The number of UDP packets received by the device.
sent	The number of UDP packets sent by the device.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.
input errors	This information is used by Dell customer support.
TCP statistics The TCP statistics are derived from RFC 793, "Transmission Control Protocol".	
active opens	The number of TCP connections opened by this device by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by this device in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Dell customer support.
active resets	The number of TCP connections this device reset by sending a TCP RESET message to the device at the other end of the connection.
passive resets	The number of TCP connections this device reset because the device at the other end of the connection sent a TCP RESET message.

TABLE 124 CLI display of IP traffic statistics (Continued)

This field...	Displays...
input errors	This information is used by Dell customer support.
in segments	The number of TCP segments received by the device.
out segments	The number of TCP segments sent by the device.
retransmission	The number of segments that this device retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.

Displaying GRE tunnel information

To display information about all GRE tunnels configured on a router, enter the following command.

```
NetIron# show gre
Total Valid GRE Tunnels : 1, GRE Session Check Enforce: FALSE
GRE tnnl 1 UP : src_ip 25.25.25.4, dst_ip 15.15.15.3
   TTL 255, TOS 0, NHT 0, MTU 1476
```

Syntax: show gre <*tunnel-number*>

The <*tunnel-number*> option allows you to limit the display to information about a specified tunnel.

Displaying GRE and manual IPv6 tunnel statistics

This section contains examples of the following **show** commands for GRE tunnel and manual IPv6 tunnel statistics:

- **show ip-tunnels**
- **show ip-tunnel** <*tunnel ID*>
- **show statistics brief tunnel** [tunnel ID]
- **show statistics tunnel** [tunnel ID]
- **show interface tunnel** <*tunnel ID*>

To see a list of the configured tunnels with some details of each tunnel, use the **show ip-tunnels** command as the following example illustrates. This example shows that one tunnel exists; it has IP tunnel statistics collection enabled; and neither GRE nor IPv6 session enforce are enabled.

```
NetIron#show ip-tunnels
# of Valid Tunnels: 1, GRE Session Enforce: FALSE, IPv6 Session Enforce: FALSE
   IP Tunnel Statistics collection Enabled
GRE tnnl 1 UP src_ip 1.1.1.4, dst_ip 1.1.1.1
   TTL 255, TOS 0, NHT 0, MTU 1476
```

Syntax: show ip tunnel <*number*>

Syntax: show ip tunnels

The output of this command contains the following type of information:

TABLE 125 Show IP tunnel display information

This field...	Displays...
Ipv6 tnnl x <*UP* | *DOWN*>	The status of the interface for manual IPv6 tunnel interface x can be one of the following: • UP – The tunnel interface is functioning properly. • DOWN – The tunnel interface is not functioning and is down.
GRE tnnl x UP or DOWN	The status of the interface for GRE tunnel interface x can be one of the following: • UP – The tunnel interface is functioning properly. • DOWN – The tunnel interface is not functioning and is down.
GRE Session Enforce	Shows whether the global GRE session enforce feature is enabled. The output is one of the following: TRUE – the feature is enabled. FALSE – the feature is disabled.
IPv6 Session Enforce	Shows whether the global IPv6 session enforce feature is enabled. The output is one of the following: TRUE – the feature is enabled. FALSE – the feature is disabled.
IP Tunnel Statistics collection	Shows whether the collection of tunnel statistics is enabled. The enable or disable is a global setting that applies to both directions of GRE and manual IPv6 tunnels (unicast and multicast).
src_ip	The tunnel source can an IPv4 address.
dst_ip	The tunnel destination can an IPv4 address.
TTL	The TTL value configured for the outer IP header. The range for TTLs is 1 – 255.
TOS	The TOS value configured for the outer IP header. The range for TOS values is 1 – 255.
NHT	The nextHop Table index value.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying brief tunnel statistics

Use the **show statistics brief tunnel** command to display the aggregate statistics for a specific tunnel or for all tunnels (in page mode). The feature combines both unicast and multicast statistics into one counter.

To display all of the brief statistics, run the **show statistics brief tunnel** command.

```
NetIron#show statistics brief tunnel
                                      Packets
Tunnel Id  Tunnel Type  [Rcv-from-tnnl    Xmit-to-tnnl]
1          GRE          586046            287497
2          GRE          0                 0
3         IPV6-Manual 0                0
```

Syntax: show statistics brief tunnel [tunnel-id]

To show the brief statistics for a particular tunnel, include the optional tunnel ID. The types of information in the output are as follows:

TABLE 126 Show IP tunnel display information

This field...	Displays...
Tunnel ID	For each tunnel displayed, the Tunnel ID indicates the tunnel for which the statistics are displayed.
Tunnel Type	The tunnel type is either GRE or manual IPv6.
Packets Rcv-from-tnnl	The number of packets that have arrived from the tunnel.
Packets Xmit-to-tnnl	The number of packets that have been sent to the tunnel.

Displaying tunnel statistics

Use the **show statistics tunnel** command to display the aggregate statistics (including unicast and multicast statistics) for a port-range for each packet processor (PPCR) for every LP module in the system. If you enter the command with no tunnel ID, the output displays statistics for all tunnels in page-mode. This command displays only the port ranges that have either unicast or multicast traffic and displays nothing if the LP does not have any non-zero counters. For this example, the LP2 is a four-port 10G card.

```
NetIron#show statistics tunnel 1
                                                    Packets
Tunnel Id  Tunnel Type  In-Port(s)     [Rcv-from-tnnl    Xmit-to-tnnl]
1          GRE          e2/1 - e2/2     586046           287497
                        e2/3 - e2/4    100340          150034
```

Syntax: show statistics tunnel [tunnel-ID]

The optional [tunnel-ID] parameter lets you specify a particular tunnel, otherwise all tunnel statistics are shown. The command output contains the following types of information.

TABLE 127 Show IP tunnel display information

This field...	Displays...
Tunnel ID	For each tunnel displayed, the Tunnel ID indicates the tunnel for which the statistics are displayed.
Tunnel Type	The tunnel type is either GRE or manual IPv6.
In-ports	The Ethernet ports traversed by the tunnel.
Packets Rcv-from-tnnl	The number of packets that have arrived from the tunnel.
Packets Xmit-to-tnnl	The number of packets that have been sent to the tunnel.

Displaying interface statistics for a tunnel

To see the interface statistics for a particular tunnel (GRE tunnel 1 in this case), use the **show interface tunnel** command, as the following illustrates.

```
NetIron#show interface tunnel 1
Tunnel 1 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source  30.30.30.1
  Tunnel destination is 20.20.20.1
  Tunnel mode gre ip
  No port name
  Internet address is: 50.50.50.4/24
  Tunnel TOS 0, Tunnel TTL 255, Tunnel MTU 1476 bytes
  Keepalive is not Enabled
  Tunnel Packet Statistics:
               Unicast Packets       Multicast Packets
    In-Port(s)   [Rcv-from-tnnl  Xmit-to-tnnl]  [Rcv-from-tnnl    Xmit-to-tnnl]
    e5/1 - e5/20   0           16511754        0             0
    e6/1 - e6/20   0           14147748        0             20195730
    e7/1 - e7/24   21493545           0        40696309      0
    e16/1 - e16/2  0            3916998        0             0
    e16/3 - e16/4  0            13476342       0
```

Syntax: show interface tunnel <*tunnel ID*>

The <*tunnel ID*> is the ID of a particular tunnel. The output contains the following types of information:

TABLE 128 Show IP tunnel display information

This field...	Displays...
Tunnel status	The tunnel and the line protocol can be • UP – The tunnel or line protocol is up and functioning properly. • DOWN – The tunnel or line protocol is down.
Tunnel source	The source IP address of the hardware.
Tunnel destination	The destination IP address of the hardware.
Tunnel mode	The tunnel mode is either GRE or manual IPv6.
Port name	The port name is displayed, or if no name has been configured, this fact is stated.
Internet address	The IP address of the ingress.
TOS	The TOS value configured for the outer IP header. The range for TOS values is 1 – 255.
TTL	The TTL value configured for the outer IP header. The range for TTLs is 1 – 255.
MTU	The setting of the IPv6 maximum transmission unit (MTU).
Keepalive	Shows the number of seconds for the keepalive option if it has been configured, otherwise it states “not Enabled.”
In-ports	The Ethernet port numbers.

TABLE 128 Show IP tunnel display information (Continued)

This field...	Displays...
Unicast Packets	On a per port basis, this column shows the number of unicast packets that have arrived from the tunnel and the number of packets that have been transmitted to the tunnel. This count includes packets for both GRE tunnels and packets for manually configured IPv6 tunnels.
Multicast Packets	On a per port basis, this column shows the number of multicast packets that have arrived from the tunnel and the packets that have been transmitted to the tunnel. This count includes packets for both GRE tunnels and packets for manually configured IPv6 tunnels.

Displaying martian addressing information

To display Martian Addressing information, use the following command.

```
NetIron# show ip martian
ip martian filtering on
0.0.0.0/8 deny
10.0.0.0/8 deny
127.0.0.0/8 deny
191.255.0.0/16 deny
192.0.0.0/24 deny
223.255.255.0/24 deny
240.0.0.0/4 deny
```

Syntax: show [vrf <*name*>**] ip martian**

You can use the **vrf** option to display martian addresses for a specific VRF.

Layer 2 Access Control Lists 20

Tthe following Layer 2 ACLs features are supported by the NetIron MLX Series devices:

- Filtering Based on Ethertype
- Filtering Based on Ethertype IPv6
- Filtering and Priority Manipulation Based on 802.1p Priority
- Binding a Layer 2 ACL Table to an Interface
- VRF ACL
- Using the Priority Option
- Using the Priority Force Option
- Using the Priority Mapping Option
- ACL Accounting

Layer-2 Access Control Lists (ACLs) filter incoming traffic based on Layer-2 MAC header fields in the Ethernet IEEE 802.3 frame. Specifically, Layer-2 ACLs filter incoming traffic based on any of the following Layer-2 fields in the MAC header:

- Source MAC address and source MAC mask
- Destination MAC address and destination MAC mask
- VLAN ID
- Ethernet type
- 802.1p

Layer-2 ACLs filter traffic at line-rate speed.

Configuration rules and notes

General considerations

- You cannot bind a Layer-2 ACL to a virtual interface.
- The Layer-2 ACL feature cannot perform SNAP and LLC encapsulation type comparisons.
- PowerConnect processes ACLs in hardware.
- You cannot edit or modify an existing Layer-2 ACL clause. If you want to change the clause, you must delete it first, then re-enter the new clause.
- You cannot add remarks to a Layer-2 ACL clause.
- When you bind a Layer-2 ACL that is not defined, it implicitly denies all traffic.

- The behavior of Layer-2 ACLs for dynamic LAG creation and deletion is that before a LAG is formed all ports which will be parts of the LAG must have the same configuration. For example, all of the ports can have no ACL, or have ACL 401 on inbound and outbound ports. After the LAG is removed, all ACL bindings (if there are any) are propagated to all of the secondary ports.
- Layer-2 inbound ACLs and Layer-2 inbound ACL-based rate limiting are not supported on Layer-3 VPNs.
- You can bind multiple rate limiting policies to a single port. However, once a matching ACL clause is found for a packet, the device does not evaluate subsequent clauses in that rate limiting ACL and subsequent rate limiting ACLs.
- Only numbered ACLs support rate limiting.

Configuration considerations for VPLS, VLL, and VLL-Local endpoints

L2 ACLs are supported on VPLS, VLL, and VLL-local endpoints with the following configuration considerations:

- First configure the port as a VPLS, VLL, or VLL-local endpoint and then bind the Layer-2 ACL on it.
- First remove the Layer-2 ACL from a VPLS, VLL, or VLL-local endpoint before removing the port from the VPLS, VLL, or VLL-local instance or corresponding VLAN.
- First remove the Layer-2 ACL from a VPLS, VLL, or VLL-local endpoint(s) before deleting the VPLS, VLL, or VLL-local instance or corresponding VLAN.
- If the VPLS, VLL, or VLL-local endpoint is a LAG port, you must first remove the Layer-2 ACL from the primary LAG port before deleting the LAG. This restriction is applicable even if you are deleting the LAG using the **force** keyword.
- If a VLL or VLL-local endpoint is a LAG port with Layer-2 ACL, you have to first remove the Layer-2 ACL from the primary LAG port before dynamically removing a port from the LAG.
- Ensure that no VPLS, VLL, or VLL-local endpoint exists with an Layer-2 ACL before entering the command: **no router mpls**.

Types of Layer-2 ACLs

Layer-2 ACLs can be numbered or named. Numbered Layer-2 ACL table IDs range from 400 to 499 and for a maximum of 100 configurable numbered Layer-2 ACL tables.

Within each Layer-2 ACL table, you can configure from 64 (default) to 256 clauses. Each clause or entry can define a set of Layer-2 parameters for filtering. Once you completely define a Layer-2 ACL table, you must bind it to the interface for filtering to take effect.

There can be up to 500 named L2 ACLs. The maximum length of a named Layer-2 ACL is 255 characters. The Layer-2 ACL name cannot begin with digits 0 to 9 to avoid confusion with the numbered L2 ACLs.

The PowerConnect evaluates traffic coming into the port against each ACL clause. Once a matching entry is found, the device either forwards or drops the traffic, depending upon the action specified for the clause. Once a matching entry is found, the device does not evaluate the traffic against subsequent clauses.

By default, if the traffic does not match any of the clauses in the ACL table, the device drops the traffic. To override this behavior, specify a “permit any any...” clause at the end of the table to match and forward all traffic not matched by the previous clauses.

NOTE
Use precaution when placing entries within the ACL table. The Layer-2 ACL feature does not attempt to resolve conflicts conflicts across multiple ACL clauses.

Creating a numbered Layer-2 ACL table

You create a numbered Layer-2 ACL table by defining a Layer-2 ACL clause.

To create a numbered Layer-2 ACL table, enter commands (clauses) such as the following at the Global CONFIG level of the CLI. Note that you can add additional clauses to the ACL table at any time by entering the command with the same table ID and different MAC parameters.

```
NetIron(config)# access-list 400 deny any any any etype arp
NetIron(config)# access-list 400 deny any any any etype ipv6
NetIron(config)# access-list 400 permit any any 100
```

This configuration creates a Layer-2 ACL with an ID of 400. When applied to an interface, this Layer-2 ACL table will deny all ARP and IPv6 traffic, and permit all other traffic in VLAN 100.

For more examples of valid Layer-2 ACL clauses, see “Filtering and priority manipulation based on 802.1p priority” on page 778.

Syntax: [no] **access-list** <num> **permit** | **deny** <src-mac> <mask> | **any** <dest-mac> <mask> | **any** [<vlan-id> | **any**] [**etype** <etype-str>] [**priority** <queue-value> | **priority-force** <queue-value> | **priority-mapping** <queue-value>]

The <num> parameter specifies the Layer-2 ACL table that the clause belongs to. The table ID can range from 400 to 499. You can define a total of 100 Layer-2 ACL tables.

The **permit** | **deny** argument determines the action to be taken when a match occurs.

The <src-mac> <mask> | **any** parameter specifies the source MAC address. You can enter a specific address and a comparison mask or the keyword any to filter on all MAC addresses. Specify the mask using Fs and zeros. For example, to match on the first two bytes of the address aabb.ccdd.eeff, use the mask ffff.0000.0000. In this case, the clause matches all source MAC addresses that contain “aabb” as the first two bytes and any values in the remaining bytes of the MAC address. If you specify any, you don’t need to specify a mask and the clause matches on all MAC addresses.

The <dest-mac> <mask> | **any** parameter specifies the destination MAC address. The syntax rules are the same as those for the <src-mac> <mask> | **any** parameter.

The optional <vlan-id> | **any** parameter specifies the vlan-id to be matched against the VLAN IDof the incoming packet. You can specify **any** to ignore the vlan-id match.

The optional **etype** <etype-str> argument specifies the Ethernet type field of the incoming packet in order for a match to occur.

The <etype-str> variable can be one of the following keywords:

- IPv4-l5 (Etype=0x0800, IPv4, HeaderLen 20 bytes)
- ARP (Etype=0x0806, IP ARP)

- IPv6 (Etype=0x86dd, IP version 6)
- ANY – specify etype any to ignore Ethernet type field match.

NOTE
Filtering based on etype value is only supported for Layer-2 inbound ACLs. It is not supported for Layer-2 outbound ACLs.

Filtering and priority manipulation based on 802.1p priority

With the Multi-Service IronWare software, Layer-2 ACL support has been provided for filtering and priority manipulation based on a packet's 802.1p priority using the following keywords.

The following priority options can be configured following the etype argument .

NOTE
The keywords **priority** and **priority-force** cannot be used together in an ACL entry.

The **priority** option assigns traffic that matches the ACL to a hardware forwarding queue. In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1q interface, this option maps the specified priority to its equivalent 802.1p (QoS) priority and marks the packet with the new 802.1p priority. This option is applicable for inbound ACLs only.

The **priority-force** option assigns packets of outgoing traffic that match the ACL to a specific hardware forwarding queue, even though the incoming packet may be assigned to another higher priority queue. This option is applicable for inbound ACLs only

The **priority-mapping** option matches on the packet's 802.1p value. This option does not change the packet's forwarding priority through the device or mark the packet. This keyword is applicable for both inbound and outbound ACLs.

The <queue value> variable specifies one of the following QoS queues for use with the priority, priority-force options

- 0 – qosp0
- 1 – qosp1
- 2 – qosp2
- 3 – qosp3
- 4 – qosp4
- 5 – qosp5
- 6 – qosp6
- 7 – qosp7

Use the **[no]** parameter to delete the Layer-2 ACL clause from the table. When all clauses are deleted from a table, the table is automatically deleted from the system.Example Numbered Layer-2 ACL Clauses

The following shows some examples of valid Layer-2 ACL clauses.

```
NetIron(config)# access-list 401 permit any any
NetIron(config)# access-list 402 permit any any 100
NetIron(config)# access-list 403 permit any any any
NetIron(config)# access-list 404 permit any any 100 etype ipv4
```

Inserting and deleting Layer-2 ACL clauses

You can make changes to the Layer-2 ACL table definitions without unbinding and rebinding the table from an interface. For example, you can add a new clause to the ACL table, delete a clause from the table, delete the ACL table, etc.

Increasing the maximum number of clauses per Layer-2 ACL table

You can increase the maximum number of clauses configurable within a Layer-2 ACL table.

To increase the maximum number of clauses per Layer-2 ACL table, enter a command such as the following at the Global CONFIG level of the CLI.

```
NetIron(config)# system-max l2-acl-table-entries 200
```

Syntax: [no] **system-max l2-acl-table-entries** <max>

NOTE
The **l2-acl-table-entries** controls the maximum number of filters supported on one Layer-2 ACL. The named Layer-2 ACL is also subject to the configuration of this **system-max** value.

The <max> parameter specifies the maximum number of clauses per Layer-2 ACL. The minimum, maximum and default values for this parameter are described in Table 4.6.

Binding a numbered Layer-2 ACL table to an interface

To enable Layer-2 ACL filtering, bind the Layer-2 ACL table to an interface. Enter a command such as the following at the Interface level of the CLI to bind an inbound Layer-2 ACL.

```
NetIron(config)# int e 4/12
NetIron(config-int-e100-4/12)# mac access-group 400 in
```

Enter a command such as the following at the Interface level of the CLI to bind an outbound Layer-2 ACL.

```
NetIron(config)# int e 4/12
NetIron(config-int-e100-4/12)# mac access-group 400 out
```

Syntax: [no] **mac access-group** <num> in | out

Filtering by MAC address

In the following example, an ACL is created that denies all traffic from the host with the MAC address 0012.3456.7890 being sent to the host with the MAC address 0011.2233.4455.

```
NetIron(config)# access-list 401 deny 0012.3456.7890 ffff.ffff.ffff
0011.2233.4455 ffff.ffff.ffff
NetIron(config)# access-list 401 permit any any
```

Using the mask, you can make the access list apply to a range of addresses. For instance if you changed the mask in the previous example from 0012.3456.7890 to ffff.ffff.fff0, all hosts with addresses from 0012.3456.7890 to 0012.3456.789f would be blocked. This configuration for this example is shown in the following.

```
NetIron(config)# access-list 401 deny 0012.3456.7890 ffff.ffff.fff0
0011.2233.4455 ffff.ffff.ffff
NetIron(config)# access-list 401 permit any any
```

The <num> parameter specifies the Layer-2 ACL table ID to bind to the interface.

Filtering broadcast traffic

To define an Layer-2 ACL that filters Broadcast traffic, enter commands such as the following.

```
NetIron(config)#access-list 401 deny any ffff.ffff.ffff ffff.ffff.ffff
NetIron(config)#access-list 401 permit any any any
```

To bind an Layer-2 ACL that filters Broadcast traffic, enter commands such as the following.

```
NetIron(config)#int eth 14/1
NetIron(config-if-e10000-14/1)#mac access-gr 401 in
```

Using the priority option

In the following example, access list 401 assigns ARP packets with any source and destination addresses from VLAN 10 to internal priority queue 5 and maps them to the 802.1p value 5 when outbound on an 802.1q interface.

```
NetIron(config)# access-list 401 permit any any 10 etype arp priority 5
```

Using the priority force option

In the following example, access list 401 assigns IPv4 packets with any source and destination addresses from VLAN 10 to the internal priority queue 6.

```
NetIron(config)# access-list 401 permit any any 10 etype ipv4-l5 priority-force 6
```

Using the priority mapping option

In the following example, access list 401 permits IPv6 packets with any source and destination addresses from VLAN 10 that have an 802.1p priority of 3.

```
NetIron(config)# access-list 401 permit any any 10 etype ipv6 priority-mapping 3
```

Creating a named Layer-2 ACL table

To create for example a named Layer-2 ACL called example_l2_acl , enter the following commands.

```
NetIron(config)#mac access-list example_l2_acl
NetIron(config-mac-nacl)#deny 0000.0000.0001 ffff.ffff.ffff any
NetIron(config-mac-nacl)#permit any 0000.0000.0002 ffff.ffff.ffff
NetIron(config-mac-nacl)#exit
```

Following is an example of how a named Layer-2 ACL “example_l2_acl” is displayed in the configuration file.

```
!
mac access-list example_l2_acl
 deny 0000.0000.0001 ffff.ffff.ffff any
 permit any 0000.0000.0002 ffff.ffff.ffff
!
```

Syntax: [no] **mac access-list** <acl_name>

Syntax: **[no] permit | deny** <src-mac> <mask> | any <dest-mac> <mask> | any [<vlan-id> | any] [etype <etype-str>] [priority <queue-value> | priority-force <queue-value> | priority-mapping <queue-value>]

Binding a named Layer-2 ACL table to an interface

Following is an example of the named Layer-2 ACL “example_l2_acl” applied to the inbound of port 2/2.

```
NetIron(config)# interface e 2/2
NetIron(config-if-e1000-2/2)#mac access-group example_l2_acl in
```

Syntax: [no] mac access-group <acl_name> <in|out>

If a Layer-2 ACL name is bound to an interface before the actual Layer-2 ACL filters are defined, the behavior will be implicit deny of all traffic. This is consistent with the behavior of other types of ACLs.

ACL accounting

Multi-Service devices monitor the number of times an ACL is used to filter incoming or outgoing traffic on an interface. The **show access-list accounting** command displays the number of “hits” or how many times ACL filters permitted or denied packets that matched the conditions of the filters. For more detailed information about ACL accounting, please refer to “ACL accounting” on page 828.

Enabling and disabling ACL accounting on NetIron MLX device

To enable ACL accounting, enter the following command in global configuration mode:

```
NetIron(config)# enable-acl-counter
```

Syntax: **[no] enable-acl-counter**

NOTE
Enabling or disabling ACL accounting affects the gathering of statistics from all ACL types (Layer-2, IPv4 and IPv6).

Displaying Layer-2 ACLs

Use the show access-list command to display Layer-2 ACL tables.

To display a Layer-2 numbered ACL table use the following command.

```
NetIron(config)#show access-list 401
```

Syntax: show access-list <number>

The <num> parameter specifies the Layer-2 ACL table ID.

To display a Layer-2 named ACL table use the following command.

```
NetIron(config)#show access-list example
```

Syntax: show access-list l2 <l2_acl_name>

The <l2_acl_name> parameter specifies the Layer-2 ACL name.

Displaying Layer-2 ACL statistics on NetIron MLX device

To display Layer 2 inbound ACL statistics on the NetIron MLX, enter commands such as the following.

```
(config-if-e10000-14/1)#show access-list acc eth 14/1 in l2
Collecting L2 ACL accounting for 400 on port 14/1  ...  Completed successfully.
L2 ACL Accounting Information:
Inbound: ACL 400
    0:   permit any any 100 etype ipv4-l5
       Hit count: (1 sec)                    0   (1 min)                    0
                  (5 min)                    0   (accum)                    0
    1:   deny any any any etype arp
       Hit count: (1 sec)                    0   (1 min)                    0
                  (5 min)                    0   (accum)                    0
```

To display Layer 2 outbound ACL statistics on the NetIron MLX, enter commands such as the following.

```
NetIron(config-if-e10000-14/1)#show access-list acc eth 14/1 out l2
Collecting L2 ACL accounting for 400 on port 14/1  ...  Completed successfully.
L2 ACL Accounting Information:
Outbound: ACL 400
    0:   permit any any 100 etype ipv4-l5
       Hit count: (1 sec)                    0   (1 min)                    0
                  (5 min)                    0   (accum)                    0
    1:   deny any any any etype arp
       Hit count: (1 sec)                    0   (1 min)                    0
                  (5 min)                    0   (accum)                    0
```

Syntax: show access-list accounting <int_type> <slot/port> <in | out> l2

Configuring ACL Deny Logging for Layer-2 inbound ACLs

Configuring ACL Deny Logging for Layer-2 ACLs requires the following:

- Enabling the Log Option

- Enabling ACL Deny Logging on an Interface

Enabling the log option

ACL Logging of Layer-2 ACLs requires that you add the **log** option to an ACL statement as shown.

```
NetIron(config)#access-list 401 deny any any any log
```

The **log** option enables logging for the Layer-2 ACL being defined.

Enabling ACL Deny Logging on an interface

The **mac access-group enable-deny-logging** command must be configured as shown on each interface that you want ACL Deny Logging for Layer-2 ACLs to function.

```
NetIron(config)# interface ethernet 5/1
NetIron(config-if-e1000-5/1)# mac access-group enable-deny-logging'
```

Syntax: [no] mac access-group enable-deny-logging [hw-drop]

The **hw-drop** option specifies that Layer-2 ACL Log packets be dropped in hardware. This is implemented to reduce the CPU load. In practice this means that the packet counts for denied traffic will only account for the first packet in each time cycle. The **no mac access-group enable-deny-logging hw-drop** command only removes the **hw-drop** keyword.

NOTE
Using this command, ACL logging can be enabled and disabled dynamically and does not require you to rebind the ACLs using the **ip rebind-acl** command

NOTE
When configuring the **mac access-group enable-deny-logging** command on VPLS, VLL, and VLL-Local endpoints, please refer to "Configuration considerations for VPLS, VLL, and VLL-Local endpoints" on page 776 for configuration guidelines.

Access Control List 21

The following ACL features are supported by the NetIron MLX Series devices.

- Access Control Lists (ACLs)
- Hardware Rule Based ACLs
- Named ACLs
- Numbered ACLs
- Standard ACLs
- Extended ACLs
- Modifying ACLs
- Deleting ACL Entries
- ACL Duplication Check
- ACL Conflict Check
- Binding IPv4 Inbound ACLs to a Management Port
- ACL CAM sharing for Inbound ACLs
- CAM sharing
- ACL Deny Logging
- ACL Accounting
- Receive ACL (RACL) Statistics
- Disabling Outbound ACLs for Switching Traffic
- Support for acl-frag-conservative
- Support for "priority" keyword in ACLs

This chapter discusses the IPv4 Access Control List (ACL) feature, which enables you to filter traffic based on the information in the IP packet header. For details on Layer 2 ACLs, refer to 20, "Layer 2 Access Control Lists". For details on IPv6 ACLs, refer to Chapter 40, "Configuring an IPv6 Access Control List".

You can use IPv4 ACLs to provide input to other features such as route maps, distribution lists, rate limiting, and BGP. When you use an ACL this way, use permit statements in the ACL to specify the traffic that you want to send to the other feature. If you use deny statements, the traffic specified by the deny statements is not supplied to the other feature. Refer to the chapters for a specific feature for information on using ACLs as input to those features.

How the PowerConnect processes ACLs

The PowerConnect processes traffic that ACLs filter in hardware. The PowerConnect creates an entry for each ACL in the Content Addressable Memory (CAM) at startup or when the ACL is created. The PowerConnect uses these CAM entries to permit or deny packets in the hardware, without sending the packets to the CPU for processing.

General configuration guidelines

Consider the following configuration guidelines:

- ACLs are supported on physical interfaces, LAG groups, and virtual routing interfaces.
- Both inbound and outbound ACLs are supported.
- You can create up to 4096 ACL entries in all the ACL configurations on the device.
- If you change the content of an ACL (add, change, or delete entries), you must remove and then reapply the ACL to all the ports that use it. Otherwise, the older version of the ACL remains in the CAM and continues to be used. You can easily re-apply ACLs using the **ip rebind-acl** <*num*> | <*name*> | **all** command. Refer to "Applying ACLs to interfaces" on page 808.
- You cannot enable any of the following features on the interface if an ACL is already applied to that interface:
  - ACL-based rate limiting
  - Policy-based routing (PBR)
  - VLAN ID Translation or Inner VLAN ID translation feature

  IP outbound and L2 outbound ACLs are mutually exclusive on NetIron MLX platform.
- Support for ACLs on MPLS VPN Endpoints – ACLs can be supported on the following endpoints:
  - IPv4 and IPv6 inbound ACLs are not supported on VPLS, VLL, or VLL-Local endpoints and vice-versa.
  - PBR route-map cannot be applied on VPLS, VLL, or VLL-Local endpoints and vice-versa.
  - The **ip access-group redirect-deny-to-inter** and **ip access-group enable-deny-logging** commands cannot be applied on VPLS, VLL, or VLL-local endpoints and vice versa.
  - IPv4 ACL-based rate limiting is not supported on VPLS and VLL endpoints.
  - Layer-2 ACLs and Layer-2 ACL-based rate limiting is not supported on Layer-3 VPNs.
  - TOS or DSCP marking using inbound ACLs is not supported on Layer-3 VPNs.
  - PBR policies are not supported on Layer-3 VPNs.

Configuration considerations for IPv4 outbound ACLs on VPLS, VLL, and VLL-Local endpoints

IPv4 outbound ACLs are supported on VPLS, VLL, and VLL-local endpoints with the following configuration considerations:

- First configure the port as a VPLS, VLL, or VLL-local endpoint and then bind the IPv4 outbound ACL on it.
- First remove the IPv4 outbound ACL from a VPLS, VLL, or VLL-local endpoint before removing the port from the VPLS, VLL, or VLL-local instance or corresponding VLAN.
- First remove the IPv4 outbound ACL from a VPLS, VLL, or VLL-local endpoint(s) before deleting the VPLS, VLL, or VLL-local instance or corresponding VLAN.
- If the VPLS, VLL, or VLL-local endpoint is a LAG port, you must first remove the IPv4 outbound ACL from the primary LAG port before deleting the LAG. This restriction is applicable even if you attempt to delete the lag using **force** keyword.

- If a VLL or VLL-local endpoint is a LAG port with a IPv4 outbound ACL, you have to first remove the IPv4 outbound ACL from the primary LAG port before dynamically removing a port from the LAG.
- Ensure that no VPLS, VLL, or VLL-local endpoint exists with an IPv4 outbound ACL before entering the command: **no router mpls**.

Disabling outbound ACLs for switching traffic

By default, when an outbound ACL is applied to a virtual interface, the PowerConnect router always filters traffic that is switched from one port to another within the same virtual routing interface. Additional commands have been added that allow you to exclude switched traffic from outbound ACL filtering. This exclusion can be configured globally or on per-port basis. This feature applies to IPv4 and IPv6 ACLs only.

All global and interface level command for disabling outbound ACLs for Switching Traffic are mutually exclusive. If the global command is configured, the interface command is not accepted. If the interface command has already been configured, configuring the global command will remove all individual port commands from the PowerConnect router's configuration.

NOTE
This feature is not recommended for MPLS interfaces.

Globally enabling outbound ACLS for switching traffic

Configuring the **acl-outbound exclude-switched-traffic** command at the general configuration level, allows you to globally exclude all switched traffic from outbound ACL filtering. This feature is configured as shown in the following.

```
NetIron(config)# acl-outbound exclude-switched-traffic ipv4
```

Syntax: [no] acl-outbound exclude-switched-traffic ipv6 | ipv4

The **ipv6** option limits the traffic excluded to IPv6 traffic only.

The **ipv4** option limits the traffic excluded to IPv4 traffic only.

The **ipv4** and **ipv6** options are mutually exclusive within the same command. If you want to configure this command to exclude both IPv4 and IPv6 traffic, you must use two separate commands.

Enabling outbound ACLS for switching traffic per port

Configuring the **if-acl-outbound exclude-switched-traffic** command at the interface configuration level, allows you to exclude all switched traffic from outbound ACL filtering on a per-port basis. With this command, one or more physical ports (for instance all ports within a VLAN) can be configured to exclude switched traffic from outbound ACL filtering.

This feature is configured as shown in the following.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e10000-3/1)# if-acl-outbound exclude-switched-traffic
```

Syntax: [no] it-acl-outbound exclude-switched-traffic [ipv6 | ipv4]

The **ipv6** option limits the traffic excluded to IPv6 traffic only.

The **ipv4** option limits the traffic excluded to IPv4 traffic only.

The **ipv4** and **ipv6** options are mutually exclusive within the same command. If you want to configure this command to exclude both IPv4 and IPv6 traffic, you must use two separate commands.

Default ACL action

The default action when no ACLs is configured on a PowerConnect is to permit all traffic. However, once you configure an ACL and apply it to a port, the default action for that port is to deny all traffic that is not explicitly permitted on the port:

- If you want to tightly control access, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.
- If you want to secure access in environments with many users, you might want to configure ACLs that consist of explicit deny entries, then add an entry to permit all access to the end of each ACL. The software permits packets that are not denied by the deny entries.

NOTE
Do not apply an empty ACL (an ACL ID without any corresponding entries) to an interface. If you accidentally do this, the software applies the default ACL action, deny all, to the interface and thus denies all traffic.

Types of IP ACLs

IP ACLs can be configured as standard or extended ACLs. A standard ACL permits or denies packets based on source IP address. An extended ACL permits or denies packets based on source and destination IP address and also based on IP protocol information.

Standard or extended ACLs can be numbered or named. Standard numbered ACLs have an ID of 1 – 99. Extended numbered ACLs are numbered 100 – 199. IDs for standard or extended ACLs can be a character string. In this document, an ACL with a string ID is called a named ACL.

ACL IDs and entries

ACLs consist of ACL IDs and ACL entries:

- **ACL ID** – An IPv4 ACL ID is a number from 1 – 99 (for a standard ACL) or 100 – 199 (for an extended ACL) or a character string. The ACL ID identifies a collection of individual ACL entries. When you apply ACL entries to an interface, you do so by applying the ACL ID that contains the ACL entries to the interface, instead of applying the individual entries to the interface. This makes applying large groups of access filters (ACL entries) to interfaces simple.
- **ACL entry** – An ACL entry are the filter commands associated with an ACL ID. These are also called “statements”. The maximum number of ACL entries you can configure is a system-wide parameter and depends on the PowerConnect you are configuring. You can configure up to the maximum number of entries in any combination in different ACLs. The total number of entries in all ACLs cannot exceed the system maximum.

You configure ACLs on a global basis, then apply them to the incoming or outgoing traffic on specific ports. You can apply only one ACL to a port's inbound traffic and only one ACL to a port's outbound traffic. The software applies the entries within an ACL in the order they appear in the ACL's configuration. As soon as a match is found, the software takes the action specified in the ACL entry (permit or deny the packet) and stops further comparison for that packet.

Enabling support for additional ACL statements

You can enable support for up to 40,960 ACL statements. To enable the PowerConnect to support 40,960 ACL entries, enter the following command at the Global CONFIG level of the CLI.

```
NetIron(config)# system-max ip-filter-sys 40960
```

Syntax: [no] system-max ip-filter-sys <*num*>

You can load ACLs dynamically by saving them in an external configuration file on flash card or TFTP server, then loading them using one of the following commands:

- **copy slot1 | slot2 running** <*from-name*>
- **ncopy slot1 | slot2** <*from-name*> **running**
- **copy tftp running-config** <*ip-addr*> <*filename*>
- **ncopy tftp** <*ip-addr*> <*from-name*> **running-config**

In this case, the ACLs are added to the existing configuration.

Configuring numbered and named ACLs

When you configure IPv4 ACLs, you can refer to the ACL by a numeric ID or by an alphanumeric name. The commands to configure numbered ACLs are different from the commands for named ACLs:

- If you refer to the ACL by a numeric ID, you can use 1 – 99 for a standard ACL or 100 – 199 for an extended ACL. This document refers to this ACL as *numbered ACL*.
- If you refer to the ACL by a name, you specify whether the ACL is a standard ACL or an extended ACL, then specify the name. This document refers to this ACL type as *named ACL*.

You can configure up to 99 standard numbered IP ACLs and 100 extended numbered IP ACLs. You also can configure up to 100 named ACLs and 500 extended named ACLs by number.

Configuring standard numbered ACLs

The following section describes how to configure standard numbered IPv4 ACLs with numeric IDs:

- For configuration information on named ACLs, refer to "Configuring standard or extended named ACLs" on page 800.
- For configuration information on extended ACLs, refer to "Configuring extended numbered ACLs" on page 791.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard ACLs. There is no limit to the number of ACL entries an ACL can contain except for the system-wide limitation. For the number of ACL entries supported on a PowerConnect, refer to "ACL IDs and entries" on page 788.

To configure a standard ACL and apply it to inbound traffic on port 1/1, enter the following commands.

```
NetIron(config)# access-list 1 deny host 209.157.22.26
NetIron(config)# access-list 1 deny 209.157.29.12
NetIron(config)# access-list 1 deny host IPHost1
NetIron(config)# access-list 1 permit any
NetIron(config)# int eth 1/1
NetIron(config-if-e10000-1/1)# ip access-group 1 in
NetIron(config)# write memory
```

The commands in this example configure an ACL to deny incoming packets from three source IP addresses from being forwarded on port 1/1. The last ACL entry in this ACL permits all packets that are not explicitly denied by the first three ACL entries.

Standard ACL syntax

Syntax: [no] access-list <*num*> **deny | permit** <*source-ip*> **|** <*hostname*> <*wildcard*>

or

Syntax: [no] access-list <*num*> **deny | permit** <*source-ip*>/<*mask-bits*> **|** <*hostname*>

Syntax: [no] access-list <*num*> **deny | permit host** <*source-ip*> **|** <*hostname*>

Syntax: [no] access-list <*num*> **deny | permit any**

Syntax: [no] ip access-group <*num*> **in**

Parameters to configure standard ACL statements

<*num*>	Enter 1 – 99 for a standard ACL.
deny | permit	Enter **deny** if the packets that match the policy are to be dropped; **permit** if they are to be forwarded.
<*source-ip*> | <*hostname*>	Specify the source IP address for the policy. Alternatively, you can specify the host name. If you want the policy to match on all source addresses, enter **any**.

NOTE
To specify the host name instead of the IP address, the host name must be configured using the **ip dns server-address...** command at the global CONFIG level of the CLI.

<*wildcard*>	Specifies the portion of the source IP host address to match against. The <*wildcard*> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <*source-ip*>. Ones mean any value matches. For example, the <*source-ip*> and <*wildcard*> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy. If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file. If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<*mask-bits*>" format. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format. **NOTE** If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show access-list** command.
host <*source-ip*> | <*hostname*>	Specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.
any	Use this parameter to configure the policy to match on all host addresses.

Parameters to bind standard ACLs to an interface

Use the **ip access-group** command to bind the ACL to an inbound interface and enter the ACL number for <*num*>.

Configuring extended numbered ACLs

This section describes how to configure extended numbered IPv4 ACLs:

- For configuration information on named ACLs, refer to "Configuring numbered and named ACLs" on page 789.
- For configuration information on standard ACLs, refer to "Configuring standard numbered ACLs" on page 789.

Extended ACLs let you permit or deny packets based on the following information:

- IP protocol

- Source IP address or host name
- Destination IP address or host name
- Source TCP or UDP port (if the IP protocol is TCP or UDP)
- Destination TCP or UDP port (if the IP protocol is TCP or UDP)

The IP protocol can be one of the following well-known names or any IP protocol number from 0 – 255:

- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Internet Gateway Routing Protocol (IGRP)
- Internet Protocol (IP)
- Open Shortest Path First (OSPF)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IP address to the website's IP address.

To configure an extended access list that blocks all Telnet traffic received on port 1/1 from IP host 209.157.22.26, create the ACL with permit and deny rules, then bind the ACL to port 1/1 using the **ip access-group** command. Enter the following commands.

```
NetIron(config)# access-list 101 deny tcp host 209.157.22.26 any eq telnet
NetIron(config)# access-list 101 permit ip any any
NetIron(config)# int eth 1/1
NetIron(config-if-e10000-1/1)# ip access-group 101 in
NetIron(config)# write memory
```

Here is another example of commands for configuring an extended ACL and applying it to an interface. These examples show many of the syntax choices.

```
NetIron(config)# access-list 102 perm icmp 209.157.22.0/24 209.157.21.0/24
NetIron(config)# access-list 102 deny igmp host rkwong 209.157.21.0/24
NetIron(config)# access-list 102 deny igrp 209.157.21.0/24 host rkwong
NetIron(config)# access-list 102 deny ip host 209.157.21.100 host 209.157.22.1
NetIron(config)# access-list 102 deny ospf any any
NetIron(config)# access-list 102 permit ip any any
```

The first entry permits ICMP traffic from hosts in the 209.157.22.x network to hosts in the 209.157.21.x network.

The second entry denies IGMP traffic from the host PowerConnect named "rkwong" to the 209.157.21.x network.

The third entry denies IGRP traffic from the 209.157.21.x network to the host PowerConnect named "**rkwong**".

The fourth entry denies all IP traffic from host 209.157.21.100 to host 209.157.22.1.

The fifth entry denies all OSPF traffic.

The sixth entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

The following commands apply ACL 102 to the incoming traffic on port 1/2 and to the outgoing traffic on port 4/3.

```
NetIron(config)# int eth 1/2
NetIron(config-if-e10000-1/2)# ip access-group 102 in
NetIron(config-if-e10000-1/2)# exit
NetIron(config)# int eth 4/3
NetIron(config-if-e10000-4/3)# ip access-group 102 out
NetIron(config)# write memory
```

Here is another example of an extended ACL.

```
NetIron(config)# access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24
NetIron(config)# access-list 103 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24
NetIron(config)# access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24 lt
telnet neq 5
NetIron(config)# access-list 103 deny udp any range 5 6 209.157.22.0/24 range 7 8
NetIron(config)# access-list 103 permit ip any any
```

The first entry in this ACL denies TCP traffic from the 209.157.21.x network to the 209.157.22.x network.

The second entry denies all FTP traffic from the 209.157.21.x network to the 209.157.22.x network.

The third entry denies TCP traffic from the 209.157.21.x network to the 209.157.22.x network, if the TCP port number of the traffic is less than the well-known TCP port number for Telnet (23), and if the TCP port is not equal to 5. Thus, TCP packets whose TCP port numbers are 5 or are greater than 23 are allowed.

The fourth entry denies UDP packets from any source to the 209.157.22.x network, if the UDP port number from the source network is 5 or 6 and the destination UDP port is 7 or 8.

The fifth entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

The following commands apply ACL 103 to the incoming and outgoing traffic on ports 2/1 and 2/2.

```
NetIron(config)# int eth 2/1
NetIron(config-if-e10000-2/1)# ip access-group 103 in
NetIron(config-if-e10000-2/1)# ip access-group 103 out
NetIron(config-if-e10000-2/1)# exit
NetIron(config)# int eth 2/2
NetIron(config-if-e10000-2/2)# ip access-group 103 in
NetIron(config-if-e10000-2/2)# ip access-group 103 out
NetIron(config)# write memory
```

Extended ACL syntax

This section presents the syntax for creating an extended IPv4 ACL and for binding the ACL to an interface. Use the **ip access-group** command in the interface level to bind the ACL to an interface.

Syntax: [no] **access-list** <*num*> **deny | permit** <*ip-protocol*>
<*source-ip*> **|** <*hostname*> <*wildcard*>
[<*operator*> <*source-tcp/udp-port*>]
<*destination-ip*> **|** <*hostname*> <*wildcard*>
[<*operator*> <*destination-tcp/udp-port*>]
[<*icmp-type*>] [**established**] [**precedence** <*name*> **|** <*num*>]

[**tos** <*number*>] [**dscp-mapping** <*number*>]
[**dscp-marking** <*number*>] | [**fragment**] [**non-fragment**]
[**option value | name | keyword**] [**priority** <*priority-value*> | **priority-force** <*priority-value*> | **priority-mapping** <*priority-value*>] [**mirror**]

Syntax: [**no**] **access-list** <*num*> **deny | permit host** <*ip-protocol*> **any any**

Syntax: [**no**] **ip access-group** <*num*> **in | out**

General parameters for extended ACLs

The following parameters apply to any extended ACL you are creating.

<*num*>	Enter 100 – 199 for an extended ACL.
deny | permit	Enter **deny** if the packets that match the policy are to be dropped; **permit** if they are to be forwarded.
<*ip-protocol*>	Indicate the type of IP packet you are filtering. You can specify a well-known name for any protocol whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI.
<*source-ip*> | <*hostname*>	Specify the source IP host for the policy. If you want the policy to match on all source addresses, enter **any**.
<*wildcard*>	Specifies the portion of the source IP host address to match against. The <*wildcard*> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <*source-ip*>. Ones mean any value matches. For example, the <*source-ip*> and <*wildcard*> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy. If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file. The IP subnet masks in CIDR format is saved in the file in "/<*mask-bits*>" format. If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show access-list** command.
<*destination-ip*> | <*hostname*>	Specify the destination IP host for the policy. If you want the policy to match on all destination addresses, enter **any.**

fragment Enter this keyword if you want to filter fragmented packets. Refer to "Enabling ACL filtering of fragmented or non-fragmented packets" on page 810.

NOTE
The **fragmented** and **non-fragmented** parameters cannot be used together in an ACL entry.

non-fragment Enter this keyword if you want to filter non-fragmented packets. Refer to "Enabling ACL filtering of fragmented or non-fragmented packets" on page 810.

NOTE
The **fragmented** and **non-fragmented** parameters cannot be used together in an ACL entry.

priority | priority-force | priority-mapping

- The **Priority** option assigns internal priority to traffic that matches the ACL. In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1q interface, this option maps the specified priority to its equivalent 802.1p (QoS) priority and marks the packet with the new 802.1p priority. This option is applicable for inbound ACLs only.
- The **Priority-force** option assigns internal priority to packets of traffic that match the ACL, even though the incoming packet may be assigned a higher priority. This option is applicable for inbound ACLs only.
- The **priority-mapping** option matches on the packet's 802.1p value. This option does not change the packet's forwarding priority through the device or mark the packet. This keyword is applicable for both inbound and outbound ACLs.

<*priority-value*> The <*priority-value*> variable specifies one of the following QoS queues for use with the **priority, priority-force** or **priority-mapping** options:

- 0 – qosp0
- 1 – qosp1
- 2 – qosp2
- 3 – qosp3
- 4 – qosp4
- 5 – qosp5
- 6 – qosp6
- 7 – qosp7

mirror Specifies mirror packets matching ACL permit clause. For more information on configuring the **acl-mirror-port** command, refer to "ACL-based inbound mirroring" on page 148.

Parameters to filter TCP or UDP packets

Use the parameters below if you want to filter traffic with the TCP or UDP packets. These parameters apply only if you entered **tcp** or **udp** for the <*ip-protocol*> parameter. For example, if you are configuring an entry for HTTP, specify **tcp eq http**.

<operator>	Specifies a comparison operator for the TCP or UDP port number. You can enter one of the following operators: • **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**. • **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**. • **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**. • **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**. • **range** – The policy applies to all TCP or UDP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range. • **established** – This operator applies only to TCP packets. If you use this operator, the policy applies to TCP packets that have the ACK (Acknowledgment) or RST (Reset) bits set on (set to "1") in the Control Bits field of the TCP packet header. Thus, the policy applies only to established TCP sessions, not to new sessions. Refer to Section 3.1, "Header Format", in RFC 793 for information about this field. **NOTE** This operator applies only to destination TCP ports, not source TCP ports.
<source-tcp/udp-port>	Enter the source TCP or UDP port number.
<destination-tcp/udp-port>	Enter the destination TCP or UDP port number.

Filtering traffic with ICMP packets

Use the following parameters if you want to filter traffic that contains ICMP packets. These parameters apply only if you specified **icmp** as the *<ip-protocol>* value.

<icmp-type> Enter one of the following values, depending on the software version the PowerConnect is running:

- any-icmp-type
- echo
- echo-reply
- information-request
- mask-reply
- mask-request
- parameter-problem
- redirect
- source-quench
- time-exceeded
- timestamp-reply
- timestamp-request
- unreachable
- *<num>*

NOTE
If the ACL is for the inbound traffic direction on a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. Refer to "Configuring numbered and named ACLs" on page 789.

precedence <*name*> | <*num*>

The precedence option for of an IP packet is set in a three-bit field following the four-bit header-length field of the packet's header.

NOTE

You can specify one of the following name or number:

- **critical** or **5** – The ACL matches packets that have the critical precedence. If you specify the option number instead of the name, specify number 5.
- **flash** or **3** – The ACL matches packets that have the flash precedence. If you specify the option number instead of the name, specify number 3.
- **flash-override** or **4** – The ACL matches packets that have the flash override precedence. If you specify the option number instead of the name, specify number 4.
- **immediate** or **2** – The ACL matches packets that have the immediate precedence. If you specify the option number instead of the name, specify number 2.
- **internet** or **6** – The ACL matches packets that have the internetwork control precedence. If you specify the option number instead of the name, specify number 6.
- **network** or **7** – The ACL matches packets that have the network control precedence. If you specify the option number instead of the name, specify number 7.
- **priority** or **1** – The ACL matches packets that have the priority precedence. If you specify the option number instead of the name, specify number 1.
- **routine** or **0** – The ACL matches packets that have the routine precedence. If you specify the option number instead of the name, specify number 0.

Using ACL QoS options to filter packets

You can filter packets based on their QoS values by entering values for the following parameters:

tos <*name*> | <*num*> Specify the IP ToS name or number.

You can specify one of the following:

- **max-reliability** or **2** – The ACL matches packets that have the maximum reliability ToS. The decimal value for this option is 2.
- **max-throughput** or **4** – The ACL matches packets that have the maximum throughput ToS. The decimal value for this option is 4.
- **min-delay** or **8** – The ACL matches packets that have the minimum delay ToS. The decimal value for this option is 8.
- **normal** or **0** – The ACL matches packets that have the normal ToS. The decimal value for this option is 0.
- <*num*> – A number from 0 – 15 that is the sum of the numeric values of the options you want. The ToS field is a four-bit field following the Precedence field in the IP header. You can specify one or more of the following. To select more than one option, enter the decimal value that is equivalent to the sum of the numeric values of all the ToS options you want to select. For example, to select the **max-reliability** and **min-delay** options, enter number 10. To select all options, select 15.

dscp-mapping <*number*> The ACL matches packets on the DSCP value. This option does not change the packet's forwarding priority through the device or mark the packet.

Parameters to mark the DSCP value in a packet

Specify the DSCP value to a packet by entering the following parameter:

Use **dscp-marking** <*number*> to mark the DSCP value in the incoming packet with the value you specify. **Dscp-marking** is not supported on outbound ACLs.

Parameters to bind standard ACLs to an interface

Use the **ip access-group** command to bind the ACL to an interface and enter the ACL number for <*num*>.

Parameters to filter IP option packets

You can filter IP Option traffic based upon the content of the IP option field in the IP header.

<*value*> You can match based upon a specified IP Option value. Values between 1 - 255 can be used.

<*keyword*> You can use the **any** keyword to match packets with IP Options or use the **ignore** keyword to match packets with or without IP Options.

NOTE
If you are configuring a filter to permit/deny rsvp or igmp packets, it will ignore IP options within the packet by default.

<*name*> You can match by using any of the following well-known options by name:

eol – Matches IP Option packets that contain the eol option.

extended-security – Matches IP Option packets that contain the extended security option.

loose-source-route – Matches IP Option packets that contain the loose source route option.

no-op – Matches IP Option packets that contain the no-op option.

record-route – Matches IP Option packets that contain the record route option.

router-alert – Matches IP Option packets that contain the router alert option.

security – Matches IP Option packets that contain the security option.

streamid – Matches IP Option packets that contain the stream id option.

strict-source-route – Matches IP Option packets that contain the strict source route option.

timestamp – Matches IP Option packets that contain the timestamp option.

Please note, the behavior of an implicit **deny ip any any** ACL filter is different than that of an explicit **deny ip any any** filter as described in the following:

- Explicit **deny ip any any** will only apply to non-option packets.
- Explicit **deny ip any any option ignore** will apply to both option and non-option packets
- Implicit **deny ip any any** will apply to both option and non-option packets

Configuring standard or extended named ACLs

The commands for configuring named ACL entries are different from the commands for configuring numbered ACL entries. The command to configure a numbered ACL is **access-list**. The command for configuring a named ACL is **ip access-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL name with one command, which places you in the configuration level for that ACL. Once you enter the configuration level for the ACL, the command syntax is the same as the syntax for numbered ACLs.

The following examples show how to configure a named standard ACL entry and a named extended ACL entry.

Configuration example for standard ACL

To configure a named standard ACL entry, enter commands such as the following.

```
NetIron(config)# ip access-list standard Net1
NetIron(config-std-nacl)# deny host 209.157.22.26
NetIron(config-std-nacl)# deny 209.157.29.12
NetIron(config-std-nacl)# deny host IPHost1
NetIron(config-std-nacl)# permit any
NetIron(config-std-nacl)# exit
NetIron(config)# int eth 1/1
NetIron(config-if-e10000-1/1)# ip access-group Net1 in
```

The commands in this example configure a standard ACL named “Net1”. The entries in this ACL deny packets from three source IP addresses from being forwarded on port 1/1. Since the implicit action for an ACL is “deny”, the last ACL entry in this ACL permits all packets that are not explicitly denied by the first three ACL entries. For an example of how to configure the same entries in a numbered ACL, refer to “Configuring standard numbered ACLs” on page 789.

Notice that the command prompt changes after you enter the ACL type and name. The “std” in the command prompt indicates that you are configuring entries for a standard ACL. For an extended ACL, this part of the command prompt is “ext“. The “nacl” indicates that are configuring a named ACL.

Syntax: **[no] ip access-list standard** <*string*> **|** <*num*>

Syntax: **[no] ip access-list standard** <*string*> **|** <*num*> **deny | permit** <*source-ip*> **|** <*hostname*> <*wildcard*>

or

Syntax: **[no] ip access-list standard** <*string*> **|** <*num*> **deny | permit** <*source-ip*>**/**<*mask-bits*> **|** <*hostname*>

Syntax: **[no] ip access-list standard** <*string*> **|** <*num*> **deny | permit host** <*source-ip*> **|** <*hostname*>

Syntax: **[no] ip access-list standard** <*string*> **|** <*num*> **deny | permit any**

Syntax: **[no] ip access-group** <*num*> **in**

The **standard** parameter indicates the ACL type.

The <*string*> parameter is the ACL name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, “ACL for Net1”). The <*num*> parameter allows you to specify an ACL number if you prefer. If you specify a number, you can specify from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

NOTE

For convenience, the software allows you to configure numbered ACLs using the syntax for named ACLs. The software also still supports the older syntax for numbered ACLs. Although the software allows both methods for configuring numbered ACLs, numbered ACLs are always formatted in the startup-config and running-config files in using the older syntax, as follows.

```
access-list 1 deny host 209.157.22.26
access-list 1 deny 209.157.22.0 0.0.0.255
access-list 1 permit any
access-list 101 deny tcp any any eq http
```

The options at the ACL configuration level and the syntax for the **ip access-group** command are the same for numbered and named ACLs and are described in "Configuring standard numbered ACLs" on page 789.

Configuration example for extended ACL

To configure a named extended ACL entry, enter commands such as the following.

```
NetIron(config)# ip access-list extended "block Telnet"
NetIron(config-ext-nacl)# deny tcp host 209.157.22.26 any eq telnet
NetIron(config-ext-nacl)# permit ip any any
NetIron(config-ext-nacl)# exit
NetIron(config)# int eth 1/1
NetIron(config-if-e10000-1/1)# ip access-group "block Telnet" in
```

NOTE

The command prompt changes after you enter the ACL type and name. The "ext" in the command prompt indicates that you are configuring entries for an extended ACL. The "nacl" indicates that are configuring a named ACL.

Syntax: **[no] ip access-list extended** <*string*> **|** <*num*>

Syntax: **[no] deny | permit** <*ip-protocol*>
<*source-ip*> **|** <*hostname*> <*wildcard*>
[<*operator*> <*source-tcp/udp-port*>]
<*destination-ip*> **|** <*hostname*> <*wildcard*>
[<*operator*> <*destination-tcp/udp-port*>]
[<*icmp-type*>] **[established] [precedence** <*name*> **|** <*num*>]
[tos <*number*>] **[dscp-mapping** <*number*>]
[fragment] [non-fragment]

Syntax: **[no] ip access-list extended** <*string*> **|** <*num*>

Syntax: **[no] deny | permit host** <*ip-protocol*> **any any**

Syntax: **[no] ip access-group** <*string*> **|** <*num*> **in I out**

The options at the ACL configuration level and the syntax for the **ip access-group** command are the same for numbered and named ACLs and are described in "Configuring extended numbered ACLs" on page 791.

Displaying ACL definitions

To display the ACLs configured on a PowerConnect, use the **show access-list** command.

Numbered ACL

For a numbered ACL, you can enter a command such as the following.

```
NetIron(config)#show access-list 99
ACL configuration:
!
Standard IP access list 10
access-list 99 deny host 10.10.10.1
access-list 99 permit any
```

Syntax: **show access-list** <*number*> **| all**

Enter the ACL's number for the <*number*> parameter:

- 1 – 99 for standard ACLs
- 100 – 199 for extended ACLs

Enter **all** if you want to display all the ACLs configured on the device.

Named ACL

For a named ACL, enter a command such as the following.

```
NetIron(config)#show access-list name entry

Standard IP access list entry
deny host 5.6.7.8
deny host 192.168.12.3
permit any
```

Syntax: show access-list name <*acl-name*>

The ACL's name for the <*acl-name*> parameter or the ACL's number for <*acl-number*>.

Modifying ACLs

When you configure any ACL, the software places the ACL entries in the ACL in the order you enter them. For example, if you enter the following entries in the order shown below, the software always applies the entries to traffic in the same order.

```
NetIron(config)#access-list 1 deny 209.157.22.0/24
NetIron(config)#access-list 1 permit 209.157.22.26
```

Thus, if a packet matches the first ACL entry in this ACL and is therefore denied, the software does not compare the packet to the remaining ACL entries. In this example, packets from host 209.157.22.26 will always be dropped, even though packets from this host match the second entry.

You can use the CLI to reorder entries within an ACL by individually removing the ACL entries and then re-adding them. To use this method, enter "**no**" followed by the command for an ACL entry, and repeat this for each ACL entry in the ACL you want to edit. After removing all the ACL entries from the ACL, re-add them.

This method works well for small ACLs such as the example above, but can be impractical for ACLs containing many entries. Therefore, the PowerConnect provides an alternative method. The alternative method lets you upload an ACL list from a TFTP server and replace the ACLs in the PowerConnect's running-config file with the uploaded list. Thus, to change an ACL, you can edit the ACL on the file server, then upload the edited ACL to the PowerConnect. You then can save the changed ACL to the PowerConnect's startup-config file.

ACL lists contain only the ACL entries themselves, not the assignments of ACLs to interfaces. You must assign the ACLs on the PowerConnect itself.

NOTE
The only commands that are valid in the ACL list are the **access-list** and **end** commands; other commands are ignored.

Modify an ACL by configuring an ACL list on a file server.

1. Use a text editor to create a new text file. When you name the file, use 8.3 format (up to eight characters in the name and up to three characters in the extension).

NOTE
Make sure the PowerConnect has network access to the TFTP server.

2. Optionally, clear the ACL entries from the ACLs you are changing by placing commands such as the following at the top of the file.

```
NetIron(config)#no access-list 1
NetIron(config)#no access-list 101
```

   When you load the ACL list into the PowerConnect, the software adds the ACL entries in the file after any entries that already exist in the same ACLs. Thus, if you intend to entirely replace an ACL, you must use the **no access-list** <*num*> command to clear the entries from the ACL before the new ones are added.

3. Place the commands to create the ACL entries into the file. The order of the separate ACLs does not matter, but the order of the entries within each ACL is important. The software applies the entries in an ACL in the order they are listed within the ACL. Here is an example of some ACL entries.

```
NetIron(config)#access-list 1 deny host 209.157.22.26
NetIron(config)#access-list 1 deny 209.157.22.0 0.0.0.255
NetIron(config)#access-list 1 permit any
NetIron(config)#access-list 101 deny tcp any any eq http
```

   The software will apply the entries in ACL 1 in the order shown and stop at the first match. Thus, if a packet is denied by one of the first three entries, the packet will not be permitted by the fourth entry, even if the packet matches the comparison values in this entry.

4. Enter the command "**end**" on a separate line at the end of the file. This command indicates to the software that the entire ACL list has been read from the file.
5. Save the text file.
6. On the PowerConnect, enter the following command at the Privileged EXEC level of the CLI:

   copy tftp running-config <*tftp-ip-addr*> <*filename*>

NOTE
This command will be unsuccessful if you place any commands other than **access-list** and **end** (at the end only) in the file. These are the only commands that are valid in a file you load using the **copy tftp running-config...** command.

7. To save the changes to the PowerConnect's startup-config file, enter the following command at the Privileged EXEC level of the CLI:

   write memory

NOTE
Do not place other commands in the file. The PowerConnect reads only the ACL information in the file and ignores other commands, including **ip access-group** commands. To assign ACLs to interfaces, use the CLI.

Adding or deleting a comment

You can add or delete comments to an IP ACL entry.

Numbered ACLs: Adding a comment

To add a comment to an ACL entry in a numbered IPv4 ACL, perform the tasks listed below.

1. Use the **show access-list** to display the entries in an ACL.

   Example

   ```
   NetIron(config-std-nacl)# show access-list 99
   Standard IP access-list 99
   deny host 1.2.4.5
   permit host 5.6.7.8
   ```

2. To add the comment "Permit all users" to filter “permit any” (the ACL remark is attached to the filter “permit any” as instructed in Step 4). Enter a command such as the following.

   ```
   NetIron(config)# access-list 99 remark Permit all users
   ```

3. Entering a **show access-list** command displays the following:

   ```
   NetIron(config-std-nacl)# show access-list 99
   Standard IP access-list 99
   deny host 1.2.4.5
   permit host 5.6.7.8

   ACL Remarks: Permit all users
   ```

4. Enter the filter “permit any”.

   Example

   ```
   NetIron (config-std-nacl)# permit any
   ```

5. Entering a **show access-list** command displays the following.

   ```
   NetIron(config-std-nacl)# show access-list 99
   Standard IP access-list 99
   deny host 1.2.4.5
   permit host 5.6.7.8
   ACL Remarks: Permit all users
   permit any
   ```

Syntax: [no] access-list <*acl-num*> **remark** <*comment-text*>

Simply entering **access-list** <*acl-num*> **remark** <*comment-text*> adds a remark to the next ACL entry you create.

The **remark** <*comment-text*> adds a comment to the ACL entry. The remark can have up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same command. Also, in order for the remark to be displayed correctly in the output of **show** commands, the comment must be entered immediately before the ACL entry it describes.

NOTE
An ACL remark is attached to each individual filter only, not to the entire ACL (ACL 199).

Complete the syntax by specifying any options you want for the ACL entry. Options you can use to configure standard or extended numbered ACLs are discussed in "Configuring standard or extended named ACLs" on page 800.

Numbered ACLs: deleting a comment

For example, if the remark "Permit all users" has been defined for ACL 99, remove the remark by entering the following command.

```
NetIron(config)# no access-list 99 remark Permit all users
```

Syntax: [no] access-list <*number*> **remark** <*comment-text*>

Named ACLs: adding a comment to a new ACL

You can add a comment to an ACL by performing the tasks listed below.

1. Use the **show access-list** command to display the contents of the ACL. For example, you may have an ACL named "entry" and a **show access-list** command shows that it has only one entry.

```
NetIron(config)# show access-list name entry
Standard IP access-list 99
deny host 1.2.4.5
```

2. Add a new entry with a remark to this named ACL by entering commands such as the following:

```
NetIron(config)#ip access-list standard entry
NetIron(config-std-nacl)# remark Deny traffic from Marketing
NetIron(config-std-nacl)# deny 5.6.7.8
```

3. Enter a **show access-list** command displays the new ACL entry with its remark.

```
NetIron(config)# show access-list name entry
Standard IP access-list entry
deny host 1.2.4.5
ACL remark: Deny traffic from Marketing
deny host 5.6.7.8
```

Syntax: [no] ip access-list standard | extended <*acl-name*>

Syntax: [no] remark <*string*>

Syntax: [no] deny <*options*> **| permit** <*options*>

The **standard | extended** parameter indicates the ACL type.

The <*acl-name*> parameter is the IPv4 ACL name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, "ACL for Net1"). The <*acl-num*> parameter allows you to specify an ACL number if you prefer. If you specify a number, enter a number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

The **remark** <*string*> adds a comment to the ACL entry that you are about to create. The comment can have up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same command. Also, in order for the remark to be displayed correctly in the output of show commands, the comment must be entered immediately before the ACL entry it describes.

Enter **deny** to deny the specified traffic or **permit** to allow the specified traffic. Complete the configuration by specifying <*options*> for the standard or extended ACL entry. Options you can use to configure standard or extended named ACLs are discussed in the section "Configuring standard or extended named ACLs" on page 800.

Named ACLs: deleting a comment

To delete a remark from a named ACL, enter the following command.

```
NetIron(config)#ip access-list standard entry
NetIron(config-std-nacl)#no remark Deny traffic from Marketing
```

Syntax: no remark <*string*>

Deleting ACL entries

Newly created ACL entries are appended to the end of the ACL list. Since ACL entries are applied to data packets in the order they appear in a list, you needed to create ACLs in the order you want them applied.

Deleting entries from numbered ACLs

If you want to delete the second entry from a numbered ACL such as ACL 99, perform the tasks listed below.

1. Display the contents of the list.

```
NetIron(config)#show access-list 99
Standard IP access-list 99
deny host 1.2.4.5
deny host 5.6.7.8
permit any
```

2. Enter the following command.

```
NetIron(config)#no access-list 99 deny host 5.6.7.8
```

3. Display the contents of the updated list.

```
NetIron(config)# show ip access-list 99
Standard IP access-list 99
deny host 1.2.4.5
permit any
```

Syntax: no access-list <*acl-number*> **<***entire-deny-or-permit-statement***>**

The <*acl-num*> parameter allows you to specify an ACL number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

You must enter the complete deny or permit statement for the <entire-deny-or-permit-statement> variable.

Complete the configuration by specifying options for the ACL entry. Options you can use to configure standard or extended numbered ACLs are discussed in "Configuring standard numbered ACLs" on page 789 and "Configuring extended numbered ACLs" on page 791.

Deleting entries from named ACLs

To delete an ACL entry from an ACL named "entry", perform the tasks listed below.

1. Enter the following command to display the contents of the ACL list.

```
NetIron#show access-list name entry
Standard IP access list entry
deny host 1.2.4.5
deny host 10.1.1.1
deny host 5.6.7.8
permit any
```

2. To delete the second ACL entry from the list, enter a command such as the following.

```
NetIron(config)#ip access-list standard entry
NetIron(config-std-nacl)#no deny host 10.1.1.1
```

3. Enter the **show access-list name entry** command to display the updated list.

```
NetIron(config)# ip show access entry all
Standard IP access list entry
deny host1.2.4.5
deny host 5.6.7.8
permit any
```

Syntax: [no] ip access-list standard | extended <*acl-name*> **|** <*acl-number*>

Syntax: no <*entire-deny-or-permit-statement***>**

The **extended | standard** parameter indicates the ACL type.

The <*acl-name*> parameter is the IPv4 ACL name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, “ACL for Net1”). The <*acl-num*> parameter allows you to specify an ACL number if you prefer. If you specify a number, enter a number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

You must enter the complete deny or permit statement for the <entire-deny-or-permit-statement> variable.

Applying ACLs to interfaces

Configuration examples in the section “Configuring numbered and named ACLs” on page 789 show that you apply ACLs to interfaces using the **ip access-group** command. This section present additional information about applying ACLs to interfaces.

Reapplying modified ACLs

If you make an ACL configuration change, you must reapply the ACLs to their interfaces to place the change into effect.

An ACL configuration change includes any of the following:

- Adding, changing, or removing an ACL or an entry in an ACL
- Changing a PBR policy
- Changing ToS-based QoS mappings

To reapply ACLs following an ACL configuration change, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# ip rebind-acl all
```

Syntax: [no] ip rebind-acl <*num*> | <*name*> **| all**

Applying ACLs to a virtual routing interface

You can apply an ACL to a virtual routing interface for both inbound and outbound traffic direction on PowerConnect. The virtual interface is used for routing between VLANs and contains all the ports within the VLAN. If the ACL is for the inbound traffic direction, you also can specify a subset of ports within the VLAN containing a specified virtual interface when assigning an ACL to that virtual interface.

Use this feature when you do not want the ACLs to apply to all the ports in the virtual interface's VLAN or when you want to streamline ACL performance for the VLAN.

To apply an ACL to a subset of ports within a virtual interface, enter commands such as the following.

```
NetIron(config)# vlan 10 name IP-subnet-vlan
NetIron(config-vlan-10)# untag ethernet 1/1 to 1/20 ethernet 2/1 to 2/12
NetIron(config-vlan-10)# router-interface ve 1
NetIron(config-vlan-10)# exit
NetIron(config)# access-list 1 deny host 209.157.22.26
NetIron(config)# access-list 1 deny 209.157.29.12
NetIron(config)# access-list 1 deny host IPHost1
NetIron(config)# access-list 1 permit any
NetIron(config)# interface ve 1
NetIron(config-vif-1)# ip access-group 1 in ethernet 1/1 ethernet 1/3 ethernet 2/1
to 2/4
```

The commands in this example configure port-based VLAN 10, add ports 1/1 – 2/12 to the VLAN, and add virtual routing interface 1 to the VLAN. The commands following the VLAN configuration commands configure ACL 1. Finally, the last two commands apply ACL 1 to a subset of the ports associated with virtual interface 1.

Syntax: [no] ip access-group <*num*> **in [ethernet** <*slot*>/<*portnum*> **| pos** <*slot*>/<*portnum*>] [<*slot*>/<*portnum*>**...] to** <*slot*>/<*portnum*>

The **ethernet** <*slot*>/<*portnum*> **| pos** <*slot*>/<*portnum*> options allow you to limit the ACL to a subset of ports within the virtual interface. You can also use the **to** <*slot*>/<*portnum*> option to specify a range of ports. A maximum of 4 port ranges are supported.

Enabling ACL duplication check

If desired, you can enable software checking for duplicate ACL entries. To do so, enter the following command at the Global CONFIG level of the CLI.

```
NetIron(config)# acl-duplication-check
```

Syntax: [no] acl-duplication-check

Enabling ACL conflict check

If desired, you can enable software checking for conflicting ACL entries. To do so, enter the following command at the Global CONFIG level of the CLI.

```
NetIron(config)# acl-conflict-check
```

Syntax: [no] acl-conflict-check

NOTE
This command only checks for conflict between ACL filters that are the same except for the permit/deny keyword.

Enabling ACL filtering of fragmented or non-fragmented packets

To define an extended IPv4 ACL to deny or permit traffic with fragmented or unfragmented packets, enter a command such as those shown in one of the methods below.

Numbered ACLs

```
NetIron(config)# access-list 111 deny ip any any fragment
NetIron(config)# int eth 1/1
NetIron(config-if-e10000-1/1)# ip access-group 111 in
NetIron(config)# write memory
```

The first line in the example defines ACL 111 to deny any fragmented packets. Other packets will be denied or permitted, based on the next filter condition.

Next, after assigning the ACL to Access Group 111, the access group is bound to port 1/1. It will be used to filter incoming traffic.

Refer to "Extended ACL syntax" on page 793 for the complete syntax for extended ACLs.

Named ACLs

```
NetIron(config)# ip access-list extended entry
NetIron(config-ext-nacl)# deny ip any any fragment

NetIron(config)# int eth 1/1
NetIron(config-if-e10000-1/1)# ip access-group entry in
NetIron(config)# write memory
```

The first line in the example defines ACL entry to deny any fragmented packets. Other packets will be denied or permitted, based on the next filter condition.

Next, after assigning the ACL to Access Group entry, the access group is bound to port 1/1. It will be used to filter incoming traffic.

Syntax: **ip access-list extended** <*acl-name*> | <*acl-num*> **deny | permit** <*ip-protocol*> <*source-ip*> | <*hostname*> <*wildcard*> [<*operator*> <*source-tcp/udp-port*>] <*destination-ip*> | <*hostname*> [<*icmp-type*> | <*num*>] <*wildcard*> [<*operator*> <*destination-tcp/udp-port*>] **[precedence** <*name*> | <*num*>] **[tos** <*name*> | <*num*>] **[fragment] | [non-fragmented]**

Enter **extended** to indicate the named ACL is an extended ACL.

The <*acl-name*> | <*acl-num*> parameter allows you to specify an IPv4 ACL name or number. If using a name, specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name, if you enclose the name in quotation marks (for example, "ACL for Net1"). The <*acl-num*> parameter allows you to specify an ACL number if you prefer. If you specify a number, enter a number from 100 – 199 for extended ACLs.

Enter the **fragment** keyword to allow the ACL to filter fragmented packets. Use the **non-fragmented** keyword to filter non-fragmented packets.

NOTE
The **fragmented** and **non-fragmented** parameters cannot be used together in an ACL entry.

Complete the configuration by specifying options for the ACL entry. Options you can use are discussed in the appropriate sections for configuring ACLs in this chapter.

Configuring the conservative ACL fragment mode

The **acl-frag-conservative** command allows you to change the operation of ACLs on fragmented packets.

When a a packet exceeds the maximum packet size, the packet is fragmented into a number of smaller packets that contain portions of the contents of the original packet. This packet flow begins with an initial packet that contains all of the Layer-3 and Layer-4 header information contained in the original packet and is following by a number of packets that contain only the Layer-3 header information. This packet flow contains all of the information contained in the original packet distributed through the packet flow into packets that are small enough to avoid the maximum packet size limit. This provides a particular problem for ACL processing. If the ACL is filtering based on Layer-4 information, the non-initial packets within the fragmented packet flow will not match the Layer-4 information even if the original packet that was fragmented would have matched the filter. Consequently, packets that the ACL was designed to filter for are not processed by the ACL.

This can be a particular problem for Deny ACLs because packets can be dropped that should be forwarded. For this reason, the conservative ACL fragment mode has been created to treat fragmented packets differently both when the **fragmented** keyword is and is not used. While under normal operation, fragmented packets are treated the same as all other packets, when the **acl-frag-conservative** command is enabled, the router only applies Layer-4 information within an ACL to non-fragmented packets and to the initial packet within a fragmented packet flow.

Layer-4 information in an ACL

An ACL entry with one or more of the following keywords is consider to have Layer-4 information:

- TCP or UDP source or destination port. (In the case of ICMP, matching based on ICMP type or code values)
- TCP SYN flag
- TCP Established flag

ACL operation with the router configured in conservative ACL fragment mode

Operation of ACLs with Fragmented packets when the router is configured in Conservative ACL Fragment mode, through use of the **acl-frag-conservative** command, can be described to follow one these four procedures:

- ACL entries with Layer-3 Information only that do not contain the **fragment** keyword
- ACL entries with Layer-3 Information only that do contain the **fragment** keyword
- ACL entries with Layer-3 and Layer-4 Information that do not contain the **fragment** keyword
- ACL entries with Layer-3 and Layer-4 Information that do contain the **fragment** keyword

Detailed operation of ACLs under each of these conditions are described as follows.

ACL entries with Layer-3 information only that do not contain the fragment keyword

In this situation, the operation of the ACL is exactly like it is during normal operation (**acl-frag-conservative** command not configured). Any packet or fragment that matches the Layer-3 information specified in the ACL will be matched as described in Table 129

TABLE 129 ACL entry with Layer-3 information only and no fragment keyword

	Packet matches AND is either a non-fragmented or the 1st packet within a fragmented packet flow	Packet matches AND is a non-initial packet within a fragmented packet flow
permit	Yes – Matches because the packet matches the Layer-3 Information in the ACL	Yes – Matches because the packet matches the Layer-3 Information in the ACL
deny	Yes – Matches because the packet matches the Layer-3 Information in the ACL	Yes – Matches because the packet matches the Layer-3 Information in the ACL

ACL entries with Layer-3 information only that do contain the fragment keyword

In this situation, any packet that is not fragmented will not match because the fragment keyword is configured in the ACL. Non-initial packets within a fragmented packet flow that contain the Layer-3 information specified in the ACL will be matched as described in Table 130.

TABLE 130 ACL entry with Layer-3 information only and fragment keyword in ACL

	Packet matches AND is either a non-fragmented or the 1st packet within a fragmented packet flow	Packet matches AND is a non-initial packet within a fragmented packet flow
permit	No – Does not match because fragment keyword is in ACL and packet is either non-fragmented or the 1st packet within a fragmented packet flow	Yes – Matches because fragment keyword is in ACL and packet is a non-initial packet within a fragmented packet flow and the packet matches the Layer-3 information in the ACL.
deny	No – Does not match because fragment keyword is in ACL and packet is either non-fragmented or the 1st packet within a fragmented packet flow	Yes – Matches because fragment keyword is in ACL and packet is a non-initial packet within a fragmented packet flow and the packet matches the Layer-3 information in the ACL.

ACL entries with Layer-3 and Layer-4 information that do not contain the fragment keyword

In this situation, any packet that is not fragmented or is the 1st packet within a fragmented packet flow and also contains the Layer-3 and Layer-4 information specified in the ACL will be matched. Packets that are non-initial packets within a fragmented packet flow and match the Layer-3 information will be matched for the permit clause because in conservative ACL fragment mode,

Layer-4 information is disregarded for non-initial packets. Also, non-initial packets within a fragmented packet flow will not be matched for the deny clause because in conservative ACL fragment mode, the deny clause is not invoked for non-initial packets within a fragmented packet flow. Refer to Table 131 for operation in this scenario

TABLE 131 ACL entry with Layer-3 and Layer-4 information and no fragment keyword in ACL

	Packet matches AND is either a non-fragmented or the 1st packet within a fragmented packet flow	Packet matches AND is a non-initial packet within a fragmented packet flow
permit	Yes – Matches because the packet matches the Layer-3 and Layer-4 Information in the ACL	Yes – Matches because the packet matches the Layer-3 Information in the ACL and in conservative mode, Layer-4 information is disregarded for non-initial packets within a fragmented packet flow
deny	Yes – Matches because the packet matches the Layer-3 and Layer-4 Information in the ACL	No – Does not match because in conservative mode, the deny clause is not invoked for non-initial packets within a fragmented packet flow.

ACL entries with Layer-3 and Layer-4 information that contains the fragment keyword

In this situation, any packet that is not fragmented or is the 1st packet within a fragmented packet flow will not be matched because the fragment keyword is specified in the ACL. Packets that are non-initial packets within a fragmented packet flow, match the **fragment** keyword and match the Layer-3 information will be matched for the permit clause because in conservative ACL fragment mode, Layer-4 information is disregarded for non-initial packets. Also, non-initial packets within a fragmented packet flow will not be matched for the deny clause because in conservative ACL fragment mode, the **deny** clause is not invoked for non-initial packets within a fragmented packet flow. Refer to Table 132 for operation in this scenario.

TABLE 132 ACL entry with Layer-3 and Layer-4 information and fragment keyword in ACL

	Packet matches AND is either a non-fragmented or the 1st packet within a fragmented packet flow	Packet matches AND is a non-initial packet within a fragmented packet flow
permit	No – Does not match because fragment keyword is in ACL and packet is either non-fragmented or the 1st packet within a fragmented packet flow	Yes – Matches because the packet matches the Layer-3 Information in the ACL and in conservative mode, Layer-4 information is disregarded for non-initial packets within a fragmented packet flow
deny	No – Does not match because fragment keyword is in ACL and packet is either non-fragmented or the 1st packet within a fragmented packet flow	No – Does not match because in conservative mode, the deny clause is not invoked for non-initial packets within a fragmented packet flow.

Configuring the conservative ACL fragment mode

The Conservative ACL Fragment Mode is configured using the **acl-frag-conservative** command as shown in the following.

```
NetIron(config)# acl-frag-conservative
```

Syntax: [no] acl-frag-conservative

Examples of ACL filtering in normal and conservative ACL fragment modes

The following examples illustrate how an ACL with the fragment keyword operates for filtering applications in both the normal and conservative mode:

- ACL Configuration Example with Fragment Keyword and Permit Clause

- ACL Configuration Example with Fragment Keyword and Deny Clause

ACL configuration example with fragment keyword and permit clause

In the following example, ACL 100 is configured to process fragmented IP packets in Normal and Conservative ACL modes as described.

```
NetIron(config)# access-list 100 permit tcp 150.1.0.0.0.0.0.255 any fragment
NetIron(config)# access-list 100 deny ip any any
```

Behavior In Normal ACL Fragment Mode – In the normal PowerConnect router mode, fragmented and non-fragmented packets will be dropped or forwarded as described in the following:

All TCP fragments (both initial and subsequent fragments) from the specified IP address, will match the first ACL entry. Because this is a **permit** ACL entry, the matching packets are forwarded.

Non-fragmented packets will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**deny**) ACL entry and consequently will be dropped.

Behavior In Conservative ACL Fragment Mode – If the PowerConnect router is configured for Conservative ACL Fragment mode using the **acl-frag-conservative** command, fragmented and non-fragmented packets will be dropped or forwarded as described in the following:

The initial fragment will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**deny**) ACL entry and consequently will be dropped.

Non-initial TCP fragments from the specified IP address, will match the first ACL entry based on Layer-3 information. Because this is a **permit** ACL entry, the matching packets are forwarded.

Non-fragmented packets will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**deny**) ACL entry and consequently will be dropped.

ACL configuration example with fragment keyword and deny clause

In the following example, ACL 101 is configured to process fragmented IP packets in Normal and Conservative ACL modes as described.

```
NetIron(config)# access-list 101 deny tcp 150.1.0.0.0.0.0.255 any fragment
NetIron(config)# access-list 101 permit ip any any
```

Behavior In Normal ACL Fragment Mode – In the normal PowerConnect router mode, fragmented and non-fragmented packets will be dropped or forwarded as described in the following:

All TCP fragments (both initial and subsequent fragments) from the specified IP address will match the first ACL entry. Because this is a **deny** ACL entry, the matching packets are dropped.

Non-fragmented packets will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**permit**) ACL entry and consequently will be forwarded.

Behavior In Conservative ACL Fragment Mode – If the PowerConnect router is configured for Conservative ACL Fragment mode using the **acl-frag-conservative** command, fragmented and non-fragmented packets will be dropped or forwarded as described in the following:

The initial fragment will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**permit**) ACL entry and consequently will be forwarded.

Non-initial TCP fragments will match the first ACL entry based on Layer-3 information. Because this is a **deny** ACL entry with Layer-3 information only, the matching packets are dropped.

Non-fragmented packets will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**permit**) ACL entry and consequently will be forwarded.

Examples of ACL-based rate limiting in normal and conservative ACL fragment modes

The following examples illustrate how an ACL with the fragment keyword operates for rate limiting applications in both the normal and conservative mode:

- ACL-based Rate Limiting Configuration Example with Fragment Keyword and Deny Clause
- ACL-based Rate Limiting Configuration Example with Fragment Keyword and Permit Clause

ACL-based rate limiting configuration example with fragment keyword and deny clause

In the following example, ACL 102 is configured to process fragmented IP packets in Normal and Conservative ACL modes as described.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e1000-3/1)# enable
NetIron(config-if-e1000-3/1)# rate-limit strict-acl
NetIron(config-if-e1000-3/1)# rate-limit input access-group 102 499992736
750000000
NetIron(config-if-e1000-3/1)# no spanning-tree
NetIron(config-if-e1000-3/1)# exit
NetIron(config)# access-list 102 deny ip any any fragment
NetIron(config)# access-list 102 permit ip any any
```

Behavior In Normal ACL Fragment Mode – In the normal PowerConnect router mode, fragmented and non-fragmented packets will be dropped or forwarded as described in the following:

All IP fragments (both initial and subsequent fragments) will match the first ACL entry. Because this is a **deny** ACL entry, and **rate-limit strict-acl** is configured, the matching packets are dropped.

Non-fragmented packets will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**permit**) ACL entry and consequently will be forwarded and rate-limited.

Behavior In Conservative ACL Fragment Mode – If the PowerConnect router is configured for Conservative ACL Fragment mode using the **acl-frag-conservative** command, fragmented and non-fragmented packets will be dropped or forwarded as described in the following:

The initial fragment will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**permit**) ACL entry and consequently will be forwarded and rate-limited.

Non-initial IP fragments will match the first ACL entry based on Layer-3 information. Because this is a **deny** ACL entry with Layer-3 information only, and **rate-limit strict-acl** is configured, the matching packets are dropped.

Non-fragmented packets will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**permit**) ACL entry and consequently will be forwarded and rate-limited.

ACL-based rate limiting configuration example with fragment keyword and permit clause

In the following example, ACL 103 is configured to process fragmented IP packets in Normal and Conservative ACL modes as described.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e1000-3/1)# enable
NetIron(config-if-e1000-3/1)# rate-limit strict-acl
NetIron(config-if-e1000-3/1)# rate-limit input access-group 103 499992736
750000000
NetIron(config-if-e1000-3/1)# no spanning-tree
NetIron(config-if-e1000-3/1)# exit
NetIron(config)# access-list 103 permit ip any any fragment
NetIron(config)# access-list 102 deny ip any any
```

Behavior In Normal ACL Fragment Mode – In the normal PowerConnect router mode, fragmented and non-fragmented packets will be dropped or forwarded as described in the following:

All IP fragments (both initial and subsequent fragments) will match the first ACL entry. Because this is a **permit** ACL entry, the matching packets are forwarded and rate-limited.

Non-fragmented packets will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**deny**) ACL entry and consequently will be dropped.

Behavior In Conservative ACL Fragment Mode – If the PowerConnect router is configured for Conservative ACL Fragment mode using the **acl-frag-conservative** commad, fragmented and non-fragmented packets will be dropped or forwarded as described in the following:

The initial fragment will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**deny**) ACL entry and consequently will be dropped.

Non-initial IP fragments will match the first ACL entry based on L3 information. Because this is a **permit** ACL entry, the matching packets are forwarded and rate-limited.

Non-fragmented packets will not match the first ACL entry because the **fragment** keyword is present. The packet will then match the second (**deny**) ACL entry and consequently will be dropped.

ACL filtering for traffic switched within a virtual routing interface

By default, a PowerConnect does not filter traffic that is switched from one port to another within the same virtual routing interface, even if an ACL is applied to the interface. You can enable the PowerConnect to filter switched traffic within a virtual routing interface. When you enable the filtering, the PowerConnect uses the ACLs applied to inbound traffic to filter traffic received by a port from another port in the same virtual routing interface. This feature does not apply to ACLs applied to outbound traffic.

To enable filtering of traffic switched within a virtual routing interface, enter the following command at the configuration level for the interface.

```
NetIron(config-vif-1)# ip access-group ve-traffic
```

Syntax: [no] ip access-group ve-traffic

Filtering and priority manipulation based on 802.1p priority

Filtering and priority manipulation based on a packet's 801.1p priority is supported in the PowerConnect devices through the following QoS options:

- **priority** – Assigns traffic that matches the ACL to a hardware forwarding queue. In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1q interface, this option maps the specified priority to its equivalent 802.1p (QoS) priority and marks the packet with the new 802.1p priority.
- **priority-force** – Assigns packets of outgoing traffic that match the ACL to a specific hardware forwarding queue, even though the incoming packet may be assigned to another queue. Specify one of the following QoS queues:
  - 0 – qosp0
  - 1 – qosp1
  - 2 – qosp2
  - 3 – qosp3
  - 4 – qosp4
  - 5 – qosp5
  - 6 – qosp6
  - 7 – qosp7

  If a packet's 802.1p value is forced to another value by its assignment to a lower value queue, it will retain that value when it is sent out through the outbound port.

  The default behavior on previous revisions of this feature was to send the packet out with the higher of two possible values: the initial 802.1p value that the packet arrived with or the new (higher) priority that the packet has been "forced" to.
- **priority-mapping** – Matches on the packet's 802.1p value. This option does not change the packet's forwarding priority through the device or mark the packet.
- **drop-precedence** – Assigns traffic that matches the ACL to a drop precedence value between 0 -3.

drop-precedence-force – This keyword applies in situations where there are conflicting priority values for packets on an Ingress port, that conflict can be resolved by performing a priority merge (the default) or by using a **force** command to direct the router to use a particular value above other values. The **drop-precedence-force** keyword specifies that if a drop precedence is applied on the port the ACL keyword will override existing or default mappings, however, if forced at the ingress port, the port value will prevail over the acl value. Assigns traffic that matches the ACL to a drop precedence value between 0 -3.

Example using the priority option (IPv4)

In the following IPv4 example, access list 100 assigns tcp packets with the source and destination addresses specified to internal priority 2 and maps them to the 802.1p value 2 when outbound.

```
NetIron(config)#access-list 100 permit tcp 100.1.1.0/24 105.23.45.0/24 priority 2
```

The **priority** parameter specifies one of the 8 internal priorities of the PowerConnect Router. Possible values are between 0 and 7. If the outgoing interface is an 802.1q interface, the packet will have its 802.1p (QoS) priority marked with the new priority defined in this ACL.

Example using the priority force option

In the following IPv4 ACL example, access list 100 assigns udp packets with the source and destination addresses specified to the internal priority 3.

```
NetIron(config)#access-list 100 permit udp 100.1.1.0/24 105.23.45.0/24
priority-force 3
```

The **priority-force** parameter specifies one of the 8 internal priorities of the PowerConnect Router. Possible values are between 0 and 7.

For limitations when using the **priority-force** parameter, please see "Configuration considerations for IPv4 outbound ACLs on VPLS, VLL, and VLL-Local endpoints" on page 786.

Example using the priority mapping option

In the following IPv4 ACL example, access list 100 permits udp packets with the source and destination addresses specified and the 802.1p priority 7.

```
NetIron(config)# access-list 100 permit udp 175.1.1.0/24 145.75.34.0/24
priority-mapping 7
```

The **priority-mapping** parameter specifies one of the eight possible 802.1p priority values. Possible values are between 0 and 7.

NOTE
When the priority configured for a physical port and the 802.1p priority of an arriving packet differ, the higher of the two priorities is used.

ICMP filtering for extended ACLs

Extended IPv4 ACL policies can be created to filter traffic based on its ICMP message type. You can either enter the description of the message type or enter its type and code IDs. All packets matching the defined ICMP message type or type number and code number are processed in hardware.

Numbered ACLs

For example, to deny the echo message type in a numbered, extended ACL, enter commands such as the following when configuring a numbered ACL.

```
NetIron(config)# access-list 109 deny icmp any any echo
```

or

```
NetIron(config)# access-list 109 deny icmp any any 8 0
```

Syntax: [no] **access-list** *<num>* **deny | permit icmp any any** *<icmp-type>* **|** *<type-number>* *<code-number>*

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

You can either enter the name of the message type for <*icmp-type*> or the message's <*type number*> and <*code number*> of the message type. Refer to Table 133 on page 819 for valid values.

Named ACLs

For example, to deny the administratively-prohibited message type in a named ACL, enter commands such as the following.

```
NetIron(config)# ip access-list extended entry
NetIron(config-ext-nacl)# deny ICMP any any administratively-prohibited
```

or

```
NetIron(config)# ip access-list extended entry
NetIron(config-ext-nacl)#deny ICMP any any 3 13
```

Syntax: **[no] ip access-list extended** <*acl-name*>
deny | permit host icmp any any <*icmp-type*> **|** <*type-number*> <*code-number*>

The **extended** parameter indicates the ACL entry is an extended ACL.

The <*acl-name*> | <*acl-num*> parameter allows you to specify an ACL name or number. If using a name, specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, "ACL for Net1"). The <*acl-num*> parameter allows you to specify an ACL number if you prefer. If you specify a number, enter a number from 100 – 199 for extended ACLs.

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

You can either use the <*icmp-type*> and enter the name of the message type or use the <*type-number*> <*code-number*> parameter to enter the type number and code number of the message. Refer to Table 133 on page 819 for valid values

TABLE 133 ICMP message types and codes

ICMP message type	Type	Code
administratively-prohibited	3	13
any-icmp-type	x	x
destination-host-prohibited	3	10
destination-host-unknown	3	7
destination-net-prohibited	3	9
destination-network-unknown	3	6
echo	8	0
echo-reply	0	0
general-parameter-problem **NOTE:** This message type indicates that required option is missing.	12	1
host-precedence-violation	3	14

TABLE 133 ICMP message types and codes

ICMP message type	Type	Code
host-redirect	5	1
host-tos-redirect	5	3
host-tos-unreachable	3	12
host-unreachable	3	1
information-reply	16	0
information-request	15	0
mask-reply	18	0
mask-request	17	0
net-redirect	5	0
net-tos-redirect	5	2
net-tos-unreachable	3	11
net-unreachable	3	0
packet-too-big	3	4
parameter-problem **NOTE:** This message includes all parameter problems	12	0
port-unreachable	3	3
precedence-cutoff	3	15
protocol-unreachable	3	2
reassembly-timeout	11	1
redirect **NOTE:**	5	x
router-advertisement	9	0
router-solicitation	10	0
source-host-isolated	3	8
source-quench	4	0
source-route-failed	3	5
time-exceeded	11	x
timestamp-reply	14	0
timestamp-request	13	0
ttl-exceeded	11	0
unreachable	3	x

Binding IPv4 inbound ACLs to a management port

You can bind a small number of IPv4 inbound ACLs to the Ethernet port on the Management Module for filtering IP traffic sent to the Management module's CPU. These ACLs are processed in software only and are not programmed in CAM. Outbound IPv4 ACLs are not supported on the Management module's Ethernet port.

The default size of IPv4 Inbound ACLs on a management port is 20 filters. This number can be set from 1 to 100 using the following command.

```
NetIron(config)# system-max mgmt-port-acl-size 100
```

Syntax: system mgmt-port-acl-size *<acls-supported>*

The *<acls-supported>* variable allows you set a maximum number of filters that are supported for the IPv4 ACL bound to the Management Module's Ethernet port.

The possible values are 1 - 100 .

The default value is 20.

NOTE
For IPv4 inbound ACL applied to management port, the user can log traffic matching both "permit" and "deny" ACL filters that have the log keyword. The command **ip access-group enable-deny-logging** is not be required to turn on logging on a management port.

IP receive ACLs

The IP receive access-control list feature (rACL) provides hardware-based filtering capability for IPv4 traffic destined for the CPU in the default VRF such as management traffic. Its purpose is to protect the management module's CPU from overloading due to large amounts of traffic sent to one of the PowerConnect router's IP interfaces. Using the rACL command, the specified ACL is applied to every interface on the PowerConnect router. This eliminates the need to add an ACL to each interface on a PowerConnect router.

The rACL feature is configured by creating an ACL to filter traffic and then specifying that ACL in the rACL command. This applies the ACL to all interfaces on the router. The destination IP address in an ACL specified by the rACL command is interpreted to apply to all interfaces in the default VRF of the router. This is implemented by programming an ACL entry in CAM that applies the ACL clause for each interface.

For example there are the following three interfaces defined on a router:

- loopback 1 = 2.2.2.2;
- ethernet 4/1 = 10.10.10.1
- virtual ethernet interface 1 = 10.10.20.1

The access list defined in the following command will act to deny ICMP traffic to each of the defined interfaces.

```
NetIron(config)# access-list 170 deny icmp host 1.1.1.1 any
```

The ACL CAM would then be programmed with the following three entries:

- deny icmp host 1.1.1.1 host 2.2.2.2
- deny icmp host 1.1.1.1 host 10.10.10.1

- deny icmp host 1.1.1.1 host 10.10.20.1

NOTE
You must rebind an rACL whenever it is changed, as described in "Rebinding an rACL definition or policy-map" on page 824, otherwise now invalid entries will still be in CAM.

NOTE
For more information on configuring the **acl-mirror-port** command for IP Receive ACLs, refer to "Specifying the destination mirror port for IP Receive ACLs" on page 152.

Configuration guidelines for IP receive ACLs

Use the following considerations when configuring IP Receive ACLs:

- **For interface level inbound IPv4 ACL or RL-ACLs**: Traffic matching rACLs will not be subject to interface-level ACL or RL-ACLs. You must take care to configure an rACL such that only management traffic matches the rACL clauses.
- **For interface level inbound L2 ACLs or RL-ACLs**: On an interface, we support either launching an IPv4 inbound or L2 inbound ACL CAM lookup, but not both. For interfaces with L2 inbound ACLs, rACL filtering will be performed by software. Therefore, only traffic permitted by L2 inbound ACL will be processed by rACLs. Note that rate-limiting using rACLs will not be applicable for such traffic.
- **VLAN ID translation or Inner VLAN ID translation**: This feature programs L2 inbound ACL CAM entries, and hence, for ports in VLAN or Inner VLAN translation group, rACL filtering is performed in software. Note that rate limiting using rACLs will not be applicable for traffic incoming on such interfaces.
- **Global DOS attack policies**: These are supported in software. The order of precedence is:
  - rACL filtering (either in hardware or software)
  - Global DOS attack policies (only in software)

NOTE
IP Receive ACLs are applicable only for line card interfaces. IP Receive ACLs are not applicable for management ethernet interfaces.

Configuring rACLs

Configuring rACLs requires the following steps:

- Configuring an rACL and Establishing the Sequence of rACL Commands
- Applying Rate Limiting on rACLs Defined Traffic
- Specifying the Maximum Number of rACL Entries
- Rebinding an rACL Definition or Policy map

You can bind multiple rACLs, up to a maximum of 199. You must however ensure that there is no explicit **permit ip any any** or **deny ip any any** clause in any of the rACLs except the last one.

NOTE
An implicit deny ip any any will be programmed at the end, after all other rACLs. This implicit clause will always be programmed to drop the matching traffic.

Configuring rACL to apply a defined ACL and establishing the sequence of rACL commands

To configure rACL to apply ACL number 101 with a sequence number of 15 to all interfaces within the default VRF for all CPU-bound traffic, enter the following command:

```
NetIron(config)# ip receive access-list 101 sequence 15
```

Syntax: [no] ip receive access-list <*acl-num*> **sequence** <*seq-num*>

The <*acl-num*> variable identifies the ACL (standard or extended) that you want to apply to all interfaces within the default VRF for all CPU-bound traffic.

The sequence <*seq-num*> option defines the sequence in which the rACL commands will be applied. Commands are applied in order of the lowest to highest sequence numbers. For example, if the following rACL commands are entered.

```
NetIron(config)# ip receive access-list 100 sequence 10
NetIron(config)# ip receive access-list 101 sequence 25
NetIron(config)# ip receive access-list 102 sequence 15
```

The effective binding of the commands will be in the following order.

```
ip receive access-list 100 sequence 10
ip receive access-list 102 sequence 15
ip receive access-list 101 sequence 25
```

Using the **[no]** option removes the rACL access list defined in the command.

Applying rate limiting on rACL defined traffic

The rACL feature allows you to apply rate limiting to CPU-bound traffic using the **policy-map** and **strict-acl** options of the **ip receive access-list** command as described in the following:

Syntax: [no] ip receive access-list <*acl-num*> **sequence** <*seq-num*> **policy-map** <*policy-map-name*>

By default, traffic matching the "permit" clause in the specified ACL is permitted and traffic matching the "deny" clause in the ACL is dropped.

When the **policy-map** option is used, traffic matching the permit clause of the specified ACL is rate-limited as defined in the policy map specified by the **<*policy-map-name*>** variable and traffic matching the "deny" clause in the ACL is permitted but not rate limited. Using the **[no]** option removes the policy map defined in the command.

When the **policy-map** option is used with the **strict-acl** option, traffic matching the permit clause of the specified ACL is rate-limited as defined in the policy map specified by the **<*policy-map-name*>** variable and traffic matching the "deny" clause in the ACL is dropped. Using the **[no]** option removes the **strict-acl** option for the rACL command defined in the command.

Specifying the maximum number of rACLs supported in CAM

You can configure the number of software ACL CAM entries available for rACLs. This is done using the following command.

```
NetIron(config)# system-max receive-cam 2048
```

Syntax: [no] system-max receive-cam <*number*>

The <*number*> variable is the maximum number of ACL CAM entries that are allowed.

Acceptable values are powers of 2 between 512 and 8192

The default value is 1024.

NOTE
You must reload the device for this command to take effect.

Rebinding an rACL definition or policy-map

If a change is made to the definition of an IP rACL or to a rate-limiting, policy map that is specified for an rACL, you must perform a rebind as shown in the following.

```
NetIron(config)# ip rebind-receive all
```

Syntax: [no] ip rebind-receive all

NOTE
If you add or delete an IP address to or from a router interface, you need to rebind the IP Receive ACLs.

Displaying accounting information for rACL statistics

To display rACL accounting information for a specific ACL, use the following command.

```
NetIron# show access-list receive accounting 102
```

Syntax: show access-list receive accounting <*acl-id*>

The <*acl-id*> variable specifies the ACL that you want to display rACL statistics for.

To clear rACL accounting information for a specific ACL, use the following command.

```
NetIron# clear access-list receive 102
```

Syntax: clear access-list receive <*acl-id*>

The <*acl-id*> variable specifies the ACL that you want to clear rACL statistics for.

Example

If you are using loopback interfaces for all BGP peering sessions, you can define an ACL that only permits BGP traffic from a specified source IP address. Where the peer source has an IP address of 1.1.1.1 and the loopback IP address on the router is 2.2.2.2, the access list command is configured as shown in the following.

```
NetIron(config)# access-list 106 permit tcp host 1.1.1.1 host 2.2.2.2 eq bgp
```

The rACL command that implements ACL 106 is configured as shown in the following.

```
NetIron(config)# ip receive access-list 106 sequence 10
```

Matching on TCP header flags for IPv4 ACLs

In this release, you can match packets for one additional TCP header flag using IPv4 ACLs. The following command implements the additional tcp parameter for IPv4 ACLs.

Syntax: [no] access-list <*num*> **permit | deny tcp any any syn**

The <*num*> parameter indicates the ACL number and must be from 1 - 99 for a standard ACL or from 100 - 199 for and extended ACL

The **tcp** parameter indicates that you are filtering the TCP header.

The **syn** parameter directs the ACL to permit or deny based upon the status of the syn flag in the TCP header. If the contents of the flag is "1" the condition is met.

ACL deny logging

The ACL Deny Logging feature records traffic flows that are denied by an ACL bound to a port. When a packet is denied by an ACL, a Syslog entry is generated and a timer is started to keep track of the packets from this packet flow. After the timer expires (default: 5 minutes), another Syslog entry is generated if there is any packet from the tracked packet flow that was denied.

ACL Deny Logging is supported for the following:

- IPv4 Inbound ACLs
- IP Receive ACLs

ACL Deny Logging is not supported for the following:

- ACL-based Rate Limiting
- Policy Based Routing
- IPv6 ACLs

Configuration notes

Carefully consider each of the following statements before configuring the ACL Deny Logging feature on your router:

- The ACL Deny Logging feature cannot be used in conjunction with the deny traffic redirection feature (command: **ip access-group redirect-deny-to-interf**). The **ip access-group redirect-deny-to-interf** command cannot be applied on VPLS, VLL, or VLL-local endpoints and vice versa. When configuring the **ip access-group redirect-deny-to-interf** command on VPLS, VLL, and VLL-Local endpoints, please refer to "Configuration considerations for IPv4 outbound ACLs on VPLS, VLL, and VLL-Local endpoints" on page 786. If you configure both features on the same interface, the ACL deny logging feature will take precedence and the deny traffic redirection will be disabled. Although disabled, deny traffic redirection will still be shown in the running configuration.
- ACL Deny Logging is a CPU-based feature. Consequently, to maintain maximum performance we recommend that you selectively enable the logging option only on the deny filters where you are interested in seeing the logs.
- ACL Deny Logging generates Syslog entries only. No SNMP traps are issued.
- The ACL Deny Logging feature is supported for inbound ACLs only.
- You can configure the maximum number of ACL session entries using the **system-max session-limit** command as described in the *Brocade MLXe and NetIron Family Configuration Guide*.
- ACL logging is applicable only for traffic matching ACL deny clauses. It is not applicable for traffic matching ACL permit clauses.

Configuring ACL deny logging for IPv4 ACLs

Configuring ACL Deny Logging for IPv4 ACLs requires the following:

- Enabling the Log Option
- Enabling ACL Deny Logging on a Interface

Enabling the log option

ACL Logging requires that you add the **log** option to an ACL statement as shown.

```
NetIron(config)#access-list 101 deny ip any any log
```

The **log** option enables logging for the ACL being defined.

The ACL or RPF logging mechanism on the Interface modules log a maximum of 256 messages per minute, and send these messages to the Management module. A rate-limiting mechanism has been added to rate-limit the number of log messages from the Interface module CPU to the Management module CPU to 5 messages per second. Because this delays the delivery of messages to the Management module, in the worst case scenario with all 256 packets arriving at the same time on the Interface module, the time values stamped by the Management module on the messages will vary by as much as 60 seconds.

Enabling ACL deny logging on an interface

The **ip access-group enable-deny-logging** command must be configured as shown on each interface that you want ACL Deny Logging to function.

```
NetIron(config)# interface ethernet 5/1
NetIron(config-if-e1000-5/1)# ip access-group enable-deny-logging
```

Syntax: [no] ip access-group enable-deny-logging [hw-drop]

NOTE
The **ip access-group enable-deny-logging** command cannot be applied on VPLS, VLL, or VLL-local endpoints and vice versa. When configuring the **ip access-group enable-deny-logging** command on VPLS, VLL, and VLL-Local endpoints, please refer to "Configuration considerations for IPv4 outbound ACLs on VPLS, VLL, and VLL-Local endpoints" on page 786.

NOTE
The command **ip access-gr enable-deny-logging** is not be required to turn on logging on management port.

The **hw-drop** option specifies that ACL Log packets be dropped in hardware. This is implemented to reduce the CPU load. In practice this means that the packet counts for denied traffic will only account for the first packet in each time cycle. The **no ip access-group enable-deny-logging hw-drop** command only removes the **hw-drop** keyword.

NOTE
Using this command, ACL logging can be enabled and disabled dynamically and does not require you to rebind the ACLs using the **ip rebind-acl** command

Configuring ACL Deny Logging for IP receive ACLs

Since ACL Logging for IP Receive ACLs applies to all CPU bound traffic it is only required that you configure the following command globally as shown.

```
NetIron(config)#ip receive access-list enable-deny-logging
```

Syntax: [no] ip receive access-list enable-deny-logging [hw-drop]

The **hw-drop** option specifies that IP Receive ACL Log packets be dropped in hardware. This is implemented to reduce the CPU load. In practice this means that the packet counts for denied traffic will only account for the first packet in each time cycle. The **no ip receive access-list enable-deny-logging hw-drop** command only removes the **hw-drop** keyword.

NOTE
Using this command, ACL logging can be enabled and disabled dynamically and does not require you to rebind the ACLs using the **ip rebind-receive-acl** command

Configuring the log timer

You can specify how long the system waits before it sends a message in the Syslog by entering a command such as the following.

```
NetIron(config)# ip access-list logging-age 2
```

Syntax: ip access-list logging-age <*minutes*>

Enter 1 – 10 minutes. The default is 5 minutes.

Support for ACL CAM sharing

For ports sharing a PPCR that have the same ACL binded to them, ACL CAM sharing only applies if all or none of the ports have ACL Deny Logging configured.

The

In the following example, ports 4/1 and 4/2 in same packet processor (PPCR) are binded with inbound ACL 101 but only port 4/2 has the **ip access-group enable-deny-logging** command configured.

```
NetIron(config)# enable-acl-cam-sharing
NetIron(config)# interface ethernet 4/1
NetIron(config-if-e1000-4/1)# ip access group 101 in
NetIron(config)# interface ethernet 4/2
NetIron(config-if-e1000-4/2)# ip access group 101 in
NetIron(config-if-e1000-4/2)# ip access-group enable-deny-logging
```

Because they do not have the same ACL Deny Logging configuration, a separate set of ACL CAM entries are programmed for ports 4/1 and 4/2.

Log example

The following examples display typical log entries where the ACL Deny Logging feature is configured.

```
[IPv4 Inbound ACL]
Dec 16 12:12:29:I:list 102 denied tcp 10.10.10.1(1024)(Ethernet 3/1
0000.0000.0010) -> 20.20.20.1(1025), 27298224 event(s)
[L2 MAC ACL]
Dec 16 12:12:29:I: MAC ACL 400 denied 1 packets on port 3/16 [SA:0000.0000.0020,
DA:0000.0000.0010, Type:IPV4-L5, VLAN:1]
```

NOTE
Log entries generated by the ACL Logging feature for POS ports do not display the source MAC address.

ACL accounting

Multi-Service devices monitor the number of times an ACL is used to filter incoming or outgoing traffic on an interface. The **show access-list accounting** command displays the number of "hits" or how many times ACL filters permitted or denied packets that matched the conditions of the filters.

NOTE
ACL accounting does not tabulate nor display the number of implicit denials by an ACL.

Counters, stored in hardware, keep track of the number of times an ACL filter is used.

The counters that are displayed on the ACL accounting report are:

- 1s – Number of hits during the last second. This counter is updated every second.
- 1m – Number of hits during the last minute. This counter is updated every one minute.
- 5m – Number of hits during the last five minutes. This counter is updated every five minutes.
- ac – Accumulated total number of hits. This counter begins when an ACL is bound to an interface and is updated every one minute. This total is updated until it is cleared.

  The accumulated total is updated every minute. For example, a minute after an ACL is bound to a port, it receives 10 hits per second and continues to receive 10 hits per second. After one minute, the accumulated total hits is 600. After 10 minutes, there will be 6000 hits.

The counters can be cleared when the device is rebooted, when an ACL is bound to or unbound from an interface, or by entering a **clear access-list** command.

ACL rate-limiting and ACL accounting

To check the availability of ACL accounting and ACL rate-limiting resources, use the **show resource** command.

```
NetIron# show resource
< . . . >
[I cntr/mtrs(1)]  2048(size),  1982(free), 03.22%(used), 0(failed)
[O cntr/mtrs(1)]  2048(size),  1984(free), 03.12%(used), 0(failed)
< . . . >
```

The above example shows only the output related to ACL rate-limiting and ACL accounting resources, and indicates that 3.22% of input resources and 3.12% of output resources have been used.

ACL Accounting interactions between L2 ACLs and IP ACLs

Dell recommends enabling ACL accounting in only one of the ACLs bound to the same port. Including ACL-accounting-enabled clauses in both ACLs can result in anomalous reporting of filtering results.

Displaying accounting statistics for all ACLs

To display a summary of the number of hits in all ACLs on a Multi-Service device, enter the following command.

```
NetIron (config)#show access-list accounting brief
Collecting ACL accounting summary for VE 1 ...  Completed successfully.
ACL Accounting Summary: (ac = accumulated since accounting started)
    Int    In ACL              Total In Hit   Out ACL             Total Out Hit
    VE 1    111                473963(1s)
                               25540391(1m)
                               87014178(5m)
                               112554569(ac)
```

The display shows the following information:

This field...	Displays...
Collecting ACL accounting summary for <*interface*>	Shows for which interfaces the ACL accounting information was collected and whether or not the collection was successful.
Int	The ID of the interface for which the statistics are being reported.
In ACL	The ID of the ACL used to filter the incoming traffic on the interface.
Total In Hit*	The number of hits from incoming traffic processed by all ACL entries (filters) in the ACL. A number is shown for each counter.
Out ACL	ID of the ACL used to filter the outgoing traffic on the interface.
Total Out Hit*	The number of hits from incoming traffic processed by all ACL entries (filters) in the ACL. A number is shown for each counter.

* The Total In Hit and Total Out Hit displays the total number of hits for all the ACL entries (or filters) in an ACL. For example, if an ACL has five entries and each entry processed matching conditions three times during the last minute, then the total Hits for the 1m counter is 15.

Syntax: show access-list accounting brief [l2 | policy-based-routing | rate-limit]

The **l2** parameter limits the display to Layer 2 ACL accounting information.

The **policy-based-routing** parameter limits the display to policy based routing accounting information.

The **rate-limit** parameter limits the display to rate limiting ACL accounting information.

IPv4 ACL accounting statistics are displayed if no option is specified.

Displaying statistics for an interface

To display statistics for an interface, enter commands such as the following.

```
NetIron (config)#show access-list accounting ve 1 in
Collecting ACL accounting for VE 1  ...  Completed successfully.
ACL Accounting Information:
Inbound: ACL 111
    1: deny tcp any any
       Hit count: (1 sec)              237000   (1 min)12502822
                  (5 min)            87014178   (accum) 99517000
    3: permit ip any any
       Hit count: (1 sec)              236961   (1 min) 13037569
                  (5 min)                   0   (accum) 13037569
    0: deny tcp 1.1.1.0 0.0.0.255 2.2.2.0 0.0.0.255
       Hit count: (1 sec)                   0   (1 min) 0
                  (5 min)                   0   (accum) 0
    2: deny udp any any
       Hit count: (1 sec)                   0   (1 min) 0
                  (5 min)                   0   (accum) 0
```

The display shows the following information:

This field...	Displays...
The IP multicast traffic snooping state	The first line of the display indicates whether IP multicast traffic snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Collecting ACL accounting summary for <*interface*>	Shows the interface included in the report and whether or not the collection was successful.
Outbound or Inbound ACL ID	Shows the direction of the traffic on the interface and the ID of the ACL used.
#	Shows the index of the ACL entry, starting with 0, followed by the permit or deny condition defined for that ACL entry. (The first entry created for an ACL is assigned the index 0. The next one created is indexed as 1, and so on.) ACL entries are arranged beginning with the entry with the highest number of hits for IPv4 ACLs. For all other options, ACL entries are displayed in order of ascending ACL filter IDs.
Hit count	Shows the number of hits for each counter.

Syntax: **show access-list accounting ethernet** [<*slot*>/<*port*> | **ve** <*ve-number*> | **pos** <*slot*>/<*portnum*>] **in** | **out** [**l2** | **policy-based-routing** | **rate-limit**]

Use **ethernet** <*slot*>/<*port*> to display a report for a physical interface.

Use **ve** <*ve-number*> to display a report for the ports that are included in a virtual routing interface. For example, if ports 1/2, 1/4, and 1/6 are all members of ve 2, the report includes information for all three ports.

Use the **in** parameter to display statistics for incoming traffic; **out** for outgoing traffic.

The **l2** parameter limits the display to Layer 2 ACL accounting information.

The **policy-based-routing** parameter limits the display to policy based routing accounting information. This option is only available for incoming traffic.

The **rate-limit** parameter limits the display to rate limiting ACL accounting information.

Clearing the ACL statistics

Statistics on the ACL account report can be cleared:

- When a software reload occurs
- When the ACL is bound to or unbound from an interface
- When you enter the **clear access-list** command, as in the following example.

```
NetIron(config)# clear access-list all
```

Syntax: **clear access-list all** | **ethernet** <*slot*>/<*port*> | **ve** <*ve-num*> | **pos** <*slot*>/<*portnum*>

Enter **all** to clear all statistics for all ACLs.

Use **pos** <*slot*>/<*port*> to clear statistics for ACLs bound to a POS port.

Use **ethernet** <*slot*>/<*port*> to clear statistics for ACLs bound to a physical port.

Use **ve** <*ve-number*> to clear statistics for all ACLs bound to ports that are members of a virtual routing interface.

Policy-Based Routing 22

Overview

The following Policy-Based Routing features are supported on the NetIron MLX Series devices.

- Policy-Based Routing (PBR)
- Next Hop VLAN Flooding
- Policy-Based Routing over a GRE Tunnel
- Setting the Output Interface to the Null Interface
- Selectively Applying Normal Routing to Packets
- Configure the Route Map

Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware. The ACLs classify the traffic. Route maps that match on the ACLs set routing attributes for the traffic.

A PBR policy specifies the next hop for traffic that matches the policy. Using standard ACLs with PBR, you can route IP packets based on their source IP address. With extended ACLs, you can route IP packets based on all of the match criteria in the extended ACL.

You can configure the PowerConnect to perform the following types of PBR based on a packet's Layer 3 and Layer 4 information:

- Select the next-hop gateway.
- Send the packet to the null interface (null0).

When a PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. If none of the policy's direct routes or next hops is available, the packets are forwarded as per the routing table.

Configuration considerations

The configuration considerations are as follows:

- A PBR policy on an interface takes precedence over a global PBR policy.
- You cannot apply PBR on a port if that port already has inbound ACLs, inbound ACL-based rate limiting, or TOS-based QoS.
- The number of route maps that you can define is limited by the system memory. When a route map is used in a PBR policy, the PBR policy uses up to 64 instances of a route map, up to 5 ACLs in a matching policy of each route map instance.
- ACLs with the **log** option configured should not be used for PBR purposes.

- PBR ignores implicit **deny ip any any** ACL entries, to ensure that for route maps that use multiple ACLs, the traffic is compared to all the ACLs. However, if an explicit **deny ip any any** is configured, traffic matching this clause will be routed normally using Layer 3 paths and will not be compared to any ACL clauses that follow this clause.
- PBR always selects the first next hop from the next hop list that is up. If a PBR policy's next hop goes down, the policy uses another next hop if available. If no next hops are available, the device routes the traffic in the normal way.
- When you change route maps or ACL definitions you must explicitly rebind the PBR policy to an interface using the **ip rebind-acl** command.
- If a PBR policy is applied globally, inbound ACLs, inbound ACL-based rate-limiting or TOS-based QoS cannot be applied to any port on the router.
- If an IPv4 option packet matches a **deny** ACL filter with the **option** keyword, the packet will be forwarded based on Layer-3 destination. If the **ignore-options** command is configured on the incoming physical port, the packet will be forwarded based on its Layer-3 destination in hardware, otherwise the packet will be sent to the CPU for software forwarding.
- If an IPv4 option packet matches a **permit** ACL filter with the option keyword, it is hardware-forwarded based on its PBR next-hop (if available). If no PBR next-hop is available, the packet is either software or hardware-forwarded (depending on whether **ignore-options** is configured), based on an IP forwarding decision.
- Policy Based Routing (PBR) currently does not support the IPv4 and IPv6 features for changing the MTU.
- Where the next hop is a GRE tunnel:
  - Packets that are larger than the tunnel's MTU are subject to IP fragmentation and PBR processing of the fragmented packets.
  - For route changes of the tunnel destination, the appropriate information is automatically propagated to the PBR feature. Depending on the configuration of the route map, a route change can change the active next hop of the PBR if it leads to the active next hop going down which triggers a new next hop selection process.

Configuring a PBR policy

To configure PBR, you define the policies using IP ACLs and route maps, then enable PBR globally or on individual interfaces. The device programs the ACLs into the Layer 4 CAM on the interfaces and routes traffic that matches the ACLs according to the instructions in the route maps.

To configure a PBR policy:

- Configure ACLs that contain the source IP addresses for the IP traffic you want to route using PBR. Refer to the section 21, "Access Control List" for details on how to configure ACLs.
- Configure a route map that matches on the ACLs and sets the route information.
- Apply the route map to an interface.

Configure the route map

After you configure the ACLs, you can configure a PBR route map that matches based on the ACLs and sets routing information in the IP traffic.

NOTE
The match and set statements described in this section are the only route-map statements supported for PBR. Other route-map statements described in the documentation apply only to the protocols with which they are described.

To configure a PBR route map, enter commands such as the following.

```
NetIron(config)# route-map test-route permit 99
NetIron(config-routemap test-route)# match ip address 99
NetIron(config-routemap test-route)# set ip next-hop 192.168.2.1
NetIron(config-routemap test-route)# exit
```

The commands in this example configure an entry in a route map named “test-route”. The **match** statement matches on IP information in ACL 99. The **set** statement changes the next-hop IP address for packets that match to 192.168.2.1.

Syntax: [no] route-map <*map-name*> **permit | deny** <*num*>

The <*map-name*> is a string of characters that names the map. Map names can be up to 32 characters in length. You can define an unlimited number of route maps on the PowerConnect, as long as system memory is available.

The **permit | deny** parameter specifies the action the PowerConnect will take if a route matches a match statement:

- If you specify a **deny** routemap instance, it is ignored and not programmed in Layer- 4 CAM.
- If you specify **permit**, the PowerConnect applies the match and set statements associated with this route map instance.

The <*num*> parameter specifies the instance of the route map you are defining. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

PBR uses up to 64 route map instances for comparison and ignores the rest.

Syntax: [no] match ip address <*ACL-num-or-name*>

The <*ACL-num*> parameter specifies a standard or extended ACL number or name.

Setting the next hop

Traffic that matches a match statement in the route map is forwarded as defined by **set** commands. Multiple **set** commands can be configured and when a match condition is met, the router works sequentially through the list of **set** commands until it finds the first “next hop” that is operational and uses it. If that “next hop” goes down, the next hop as defined in a **set** command is chosen and if all next hop interfaces in the list are down, the packet is routed as determined in the IP Route Table. If a next hop interface that was down comes back up, the next hop selection process begins again and restarts its selection process from the top of the list.

Options for setting the next hop are described in the following:

- Setting the Next Hop to an IP Address
- Setting the Next Hop to a GRE Tunnel
- Setting the Next Hop to a Null Interface
- Setting the Next Hop to an LSP
- Setting the Next Hop to VLAN Flooding

Setting the next hop to an IP address

You can set the next hop to an IP address as shown in the following.

```
NetIron(config)# route-map net10web permit 101
NetIron(config-routemap net10web)# match ip address 101
NetIron(config-routemap net10web)# set ip next-hop 1.1.1.1
```

Syntax: [no] set ip next-hop <*ip-address*>

The <*ip-address*> variable specifies the IP address of the next-hop IP address for traffic that matches a match statement in the route map.

NOTE
If the IP address used in this command is the IP address of a configured GRE tunnel, the configuration will still be accepted but the next-hop slection will never choose this next-hop so it will not become active. If you want to set the next hop using a GRE tunnel, you must use the **set next-hop-ip-tunnel** command.

Setting the next hop to a GRE tunnel

You can set the next hop to a GRE Tunnel as shown in the following:

Syntax: [no] set next-hop-ip-tunnel <*tunnel-id*>

This command sets the next hop to the GRE tunnel identifed by the <*tunnel-id*> variable. Only GRE tunnels are supported by this command. The system will verify if a valid GRE tunnel with the specifed <*tunnel-id*> variable exists. If the <*tunnel-id*> variable points to a tunnel other than a GRE tunnel or to a non-existent tunnel, the configuration will be rejected.

Values for the <*tunnel-id*> variable can be from 1 to the maximum number of allowed Tunnel IDs in the system, The maximum number of Tunnel IDs allowed is set using the **system-max ip-tunnels** command.

For an example of a configuration using this command, refer to "Setting the next hop to a GRE tunnel" on page 840.

Setting the next hop to a Null0 interface

Sending traffic to a Null0 Interface drops the traffic. You can set the next hop to a Null0 interface as shown in the following.

```
NetIron(config)# route-map file-13 permit 56
NetIron(config-routemap file-13)# match ip address 56
NetIron(config-routemap file-13)# set interface null0
```

Syntax: [no] set interface null0

Setting the next hop to an LSP

You can set the next hop to an LSP as shown in the following.

```
NetIron(config)# route-map pbrmap permit 10
NetIron(config-routemap pbrmap)# match ip address 101
NetIron(config-routemap pbrmap)# set next-hop-lsp t3
```

Syntax: [no] set next-hop-lsp <*lsp-name*>

This command allows you to forward matching traffic to an RSVP -signalled LSP that is specified by the <*lsp-name*> variable.

Setting next hop VLAN flooding

This feature supports the ability to use PBR to forward traffic to a VLAN through use of a new "set" command. Using this feature, matched traffic can be flooded on all ports of the VLAN except the incoming physical port. Any PBR policy that contains the **set next-flood-vlan** statement applies to both routed and switched traffic. This means that if any instance in a PBR route-map contains the **set next-flood-vlan** statement, all instances of that route-map will be applied to both routed and switched traffic.

The following example floods all traffic matched from ACL 101 on all ports of VLAN 10 except the incoming physical port.

```
NetIron(config)# access-list 101 permit ip any any
NetIron(config)# route-map calea permit 10
NetIron(config-routemap calea)# match ip address 101
NetIron(config-routemap calea)# set next-flood-vlan 10
NetIron(config-routemap calea)# exit
```

Syntax: [no] set next-flood-vlan <*vlan-id*> **[outgoing da** <*mac-address*> **]**

If the VLAN specified by the <*vlan-id*> variable is not configured, the PBR route-map set statement will fall through to the next configured set statement. If no valid next-hop is available, the packet is forwarded as per L2/L3 forwarding decision. If the VLAN specified by the <*vlan-id*> variable has no valid outgoing ports, (such as when all ports in the VLAN are down or when the VLAN is empty) matching packets will be dropped.

The **outgoing da** option directs the router to send packets flooded to the ports on the VLAN to carry the destination MAC address specified in the <*mac-address*> variable.

If the destination MAC address is not set using the **outgoing da** option, the destination address is set as described in Table 134.

TABLE 134 Destination address on VLAN flooded packets

Incoming port	Outgoing port	Routed traffic	Switched traffic
Ethernet	Ethernet	Replaced Destination Address	Original Destination Address from Incoming Packet
POS	Ethernet	Replaced Destination Address	N/A

The **no set next-hop-flood-vlan** <*vlan-id*> **outgoing-da** <*mac-address*> command deletes only the outgoing-da option from the set statement. It does not delete the set statement itself. To delete the set statement, the user would have to specify the **no set next-hop-flood-vlan** <*vlan-id*> command .

In the case of traffic incoming on MPLS uplink, PBR to VLAN flooding is only supported for IPv4 traffic, and not for MPLS traffic.

Enabling PBR

After you configure the ACLs and route map entries, you can enable PBR globally, on individual interfaces, or both as described in this section. To enable PBR, you apply a route map you have configured for PBR globally or locally.

Enabling PBR globally

To enable PBR globally, enter a command such as the following at the global CONFIG level.

```
NetIron(config)# ip policy route-map test-route
```

This command applies a route map named “test-route” to all interfaces on the device for PBR.

Syntax: [no] ip policy route-map <*map-name*>

Enabling PBR locally

To enable PBR locally, enter commands such as the following.

```
NetIron(config)# interface ve 1
NetIron(config-vif-1)# ip policy route-map test-route
```

The commands in this example change the CLI to the Interface level for virtual interface 1, then apply the “test-route” route map to the interface. You can apply a PBR route map to Ethernet ports, POS ports, or virtual interfaces.

Syntax: [no] ip policy route-map <*map-name*>

Enter the name of the route map you want to use for the route-map <*map-name*> parameter.

Configuration examples

This section presents configuration examples for:

- “Basic example” on page 838
- “Setting the next hop” on page 838
- “Setting the output interface to the null interface” on page 840
- “Selectively applying normal routing to packets” on page 841

Basic example

The following commands configure and apply a PBR policy that routes HTTP traffic received on virtual routing interface 1 from the 10.10.10.x/24 network to 5.5.5.x/24 through next-hop IP address 1.1.1.1 or, if 1.1.1.x is unavailable, through 2.2.2.1.

```
NetIron(config)# access-list 101 permit tcp 10.10.10.0 0.0.0.255 eq http 5.5.5.0
NetIron(config)# route-map net10web permit 101
NetIron(config-routemap net10web)# match ip address 101
NetIron(config-routemap net10web)# set ip next-hop 1.1.1.1
NetIron(config-routemap net10web)# set ip next-hop 2.2.2.1
NetIron(config-routemap net10web)# exit
NetIron(config)# vlan 10
NetIron(config-vlan-10)# tagged ethernet 1/1 to 1/4

NetIron(config-vlan-10)# router-interface ve 1
NetIron(config)# interface ve 1
NetIron(config-vif-1)# ip policy route-map net10web
```

Setting the next hop

The following commands configure the PowerConnect to apply PBR to traffic from IP subnets 209.157.23.x, 209.157.24.x, and 209.157.25.x. In this example, route maps specify the next-hop gateway for packets from each of these subnets:

- Packets from 209.157.23.x are sent to 192.168.2.1.

- Packets from 209.157.24.x are sent to 192.168.2.2.
- Packets from 209.157.25.x are sent to 192.168.2.3.

The following commands configure three standard ACLs. Each ACL contains one of the ACLs listed above. Make sure you specify **permit** instead of deny in the ACLs, so that the PowerConnect permits the traffic that matches the ACLs to be further evaluated by the route map. If you specify **deny**, the traffic that matches the **deny** statements are routed normally. Notice that these ACLs specify **any** for the destination address.

```
NetIron(config)# access-list 50 permit 209.157.23.0 0.0.0.255
NetIron(config)# access-list 51 permit 209.157.24.0 0.0.0.255
NetIron(config)# access-list 52 permit 209.157.25.0 0.0.0.255
```

The following commands set an RSVP-signalled LSP as the next hop.

```
NetIron(config)# access-list 101 permit tcp any any
NetIron(config)# access-list 101 deny ip any any
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface ethernet 6/1
NetIron(config-mpls)# lsp t3
NetIron(config-mpls-lsp-t3)# to 30.1.1.1
NetIron(config-mpls-lsp-t3)# enable
NetIron(config)# route-map pbrmap permit 10
NetIron(config-routemap pbrmap)# match ip address 101
NetIron(config-routemap pbrmap)# set next-hop-lsp t3
```

The following commands configure three entries in a route map called “test-route”. The first entry (permit 50) matches on the IP address information in ACL 50 above. For IP traffic from subnet 209.157.23.0/24, this route map entry sets the next-hop IP address to 192.168.2.1.

```
NetIron(config)# route-map test-route permit 50
NetIron(config-routemap test-route)# match ip address 50
NetIron(config-routemap test-route)# set ip next-hop 192.168.2.1
NetIron(config-routemap test-route)# exit
```

The following commands configure the second entry in the route map. This entry (permit 51) matches on the IP address information in ACL 51 above. For IP traffic from subnet 209.157.24.0/24, this route map entry sets the next-hop IP address to 192.168.2.2.

```
NetIron(config)# route-map test-route permit 51
NetIron(config-routemap test-route)# match ip address 51
NetIron(config-routemap test-route)# set ip next-hop 192.168.2.2
NetIron(config-routemap test-route)# exit
```

The following commands configure the third entry in the test-route route map. This entry (permit 52) matches on the IP address information in ACL 52 above. For IP traffic from subnet 209.157.25.0/24, this route map entry sets the next-hop IP address to 192.168.2.3.

```
NetIron(config)# route-map test-route permit 52
NetIron(config-routemap test-route)# match ip address 52
NetIron(config-routemap test-route)# set ip next-hop 192.168.2.3
NetIron(config-routemap test-route)# exit
```

The following command enables PBR by globally applying the test-route route map to all interfaces.

```
NetIron(config)# ip policy route-map test-route
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the three source subnets identified in ACLs 50, 51, and 52, then apply route map test-route the interface.

```
NetIron(config)# interface ve 1
NetIron(config-vif-1)# ip address 209.157.23.1/24
NetIron(config-vif-1)# ip address 209.157.24.1/24
NetIron(config-vif-1)# ip address 209.157.25.1/24
NetIron(config-vif-1)# ip policy route-map test-route
```

Setting the next hop to a GRE tunnel

This section describes how to configure a PowerConnect to apply PBR to traffic on port 1/4 from subnets 11.12.13.x and 14.15.16.x. Packets from these subnets are then sent to a next hop that is a GRE tunnel. In this configuration, two GRE tunnels are configured to provide redundancy. If the first tunnel in the configuration (Tunnel 1) is down, traffic will be routed to the second tunnel (Tunnel 2). In situations where both tunnels are down, traffic from the subnets will be routed as directed from the IP route table.

```
NetIron(config)# interface ethernet 1/4
NetIron(config-if-e1000-1/1)# ip policy route-map test1
NetIron(config-if-e1000-1/1)# exit
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)# tunnel mode gre ip
NetIron(config-tnif-1)# tunnel source ethernet 1/2
NetIron(config-tnif-1)# tunnel destination 36.0.8.108
NetIron(config-tnif-1)# ip address 10.10.3.2/24
NetIron(config-tnif-1)# exit

NetIron(config)# interface tunnel 2
NetIron(config-tnif-2)# tunnel mode gre ip
NetIron(config-tnif-2)# tunnel source ethernet 2/2
NetIron(config-tnif-2)# tunnel destination 36.0.9.108
NetIron(config-tnif-2)# ip address 10.10.4.2/24
NetIron(config-tnif-2)# exit
NetIron(config)# access-list 99 permit 11.12.13.0 0.0.0.255
NetIron(config)# access-list 99 permit 14.15.16.0 0.0.0.255

NetIron(config)# route-map test1 permit 5
NetIron(config-routemap test1)# match ip address 99
NetIron(config-routemap test1)# set next-hop-ip-tunnel 1
NetIron(config-routemap test1)# set next-hop-ip-tunnel 2
```

Setting the output interface to the null interface

The following commands configure a PBR to send all traffic from 209.168.1.204 to the null interface, thus dropping the traffic instead of forwarding it.

```
NetIron(config)# access-list 56 permit 209.168.1.204 0.0.0.0
```

The following commands configure an entry in a route map called “file-13”. The first entry (permit 56) matches on the IP address information in ACL 56 above. For IP traffic from the host 209.168.1.204/32, this route map entry sends the traffic to the null interface instead of forwarding it, thus sparing the rest of the network the unwanted traffic.

```
NetIron(config)# route-map file-13 permit 56
NetIron(config-routemap file-13)# match ip address 56
NetIron(config-routemap file-13)# set interface null0
NetIron(config-routemap file-13)# exit
```

The following command enables PBR by globally applying the route map to all interfaces.

```
NetIron(config)# ip policy route-map file-13
```

Alternatively, you can enable the PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the source subnet identified in ACL 56, then apply route map file-13 to the interface.

```
NetIron(config)# interface ethernet 3/11
NetIron(config-if-e10000-3/11)# ip address 192.168.1.204/32
NetIron(config-if-e10000-3/11)# ip policy route-map file-13
```

Selectively applying normal routing to packets

This example demonstrates how to configure PBR to route all TCP traffic from a host normally while routing all other traffic from the same host through the PBR next hop. In this example, the IP address of the host is 192.168.2.2.

To route TCP traffic from 192.168.2.2 normally, configure a **deny** ACL clause and define it as a **permit route-map** entry as shown in the following.

```
NetIron(config)# access-list 112 deny tcp host 192.168.2.2 any
NetIron(config)# access-list 112 permit ip host 192.168.2.2 any
NetIron(config)# route-map mymap2 permit 10
NetIron(config-routemap mymap2)# match ip address 112
NetIron(config-routemap mymap2)# set ip next-hop 11.1.1.2
```

LAG formation

When a LAG is formed, all ports must have the same PBR configuration before deployment, during deployment the configuration on the primary port is replicated to all ports and on undeployment each port inherits the same PBR configuration.

Configuring RIP 23

Overview

The following displays the RIP features supported by Brocade NetIron XMR Series.

- RIP V1
- RIP V1 compatible with V2
- RIP Version 2 (the default)
- Administrative Distances
- Redistribution
- Route Learning and Advertising Parameters
- Changing the Route Loop Prevention Method
- Suppressing RIP Route Advertisement on a VRRP or VRRPE Backup Interface
- RIP Timers
- RIP Filters

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a ***distance vector*** (a number representing distance) to measure the cost of a given route. The ***cost*** is a distance vector because the cost often is equivalent to the number of router hops between the device and the destination network.

A device can receive multiple paths to a destination. The software evaluates the paths, selects the best path, and saves the path in the IP route table as the route to the destination. Typically, the best path is the path with the fewest hops. A hop is another router through which packets must travel to reach the destination. If the device receives a RIP update from another router that contains a path with fewer hops than the path stored in the device's route table, the device replaces the older route with the newer one. The device then includes the new path in the updates it sends to other RIP routers, including device.

RIP routers, including the PowerConnect, also can modify a route's cost, generally by adding to it, to bias the selection of a route for a given destination. In this case, the actual number of router hops may be the same, but the route has an administratively higher cost and is thus less likely to be used than other, lower-cost routes.

A RIP route can have a maximum cost of 15. Any destination with a higher cost is considered unreachable. Although limiting to larger networks, the low maximum hop count prevents endless loops in the network.

The device supports the following RIP versions:

- Version 1
- V1 compatible with V2
- Version 2 (the default)

RIP parameters and defaults

The following tables list the RIP parameters, their default values, and where to find configuration information.

RIP global parameters

Table 135 lists the global RIP parameters and their default values, and indicates where you can find configuration information.

TABLE 135 RIP global parameters

Parameter	Description	Default	See page...
RIP state	The global state of the protocol **NOTE:** You also must enable the protocol on individual interfaces. Globally enabling the protocol does not allow interfaces to send and receive RIP information. Refer to Table 136 on page 845.	Disabled	page 845
Administrative distance	The administrative distance is a numeric value assigned to each type of route on the router. When the router is selecting from among multiple routes (sometimes of different origins) to the same destination, the router compares the administrative distances of the routes and selects the route with the lowest administrative distance.	120	page 846
Redistribution	RIP can redistribute routes from other routing protocols such as OSPF and BGP4 into RIP. A redistributed route is one that a router learns through another protocol, then distributes into RIP.	Disabled	page 847
Redistribution metric	RIP assigns a RIP metric (cost) to each external route redistributed from another routing protocol into RIP. This parameter applies to routes that are redistributed from other protocols into RIP.	1 (one)	page 848
Learning default routes	The router can learn default routes from its RIP neighbors. **NOTE:** You also can enable or disable this parameter on an individual interface basis. Refer to Table 136 on page 845.	Disabled	page 849
Advertising and learning with specific neighbors	The device learns and advertises RIP routes with all its neighbors by default. You can prevent the device from advertising routes to specific neighbors or learning routes from specific neighbors.	Learning and advertising permitted for all neighbors	page 849

RIP interface parameters

Table 136 lists the interface-level RIP parameters and their default values, and indicates where you can find configuration information.

TABLE 136 RIP interface parameters

Parameter	Description	Default	See page...
RIP state and version	The state of the protocol and the version that is supported on the interface. The version can be one of the following: • Version 1 only • Version 2 only • Version 1, but also compatible with version 2 **NOTE:** : You also must enable RIP globally.	Disabled	page 845
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	page 846
Learning default routes	Locally overrides the global setting. Refer to Table 135 on page 844.	Disabled	page 849
Loop prevention	The method a router uses to prevent routing loops caused by advertising a route on the same interface as the one on which the router learned the route. • Split horizon – The router does not advertise a route on the same interface as the one on which the router learned the route. • Poison reverse – The router assigns a cost of 16 ("infinite" or "unreachable") to a route before advertising it on the same interface as the one on which the router learned the route.	Split Horizon **NOTE:** Disabling poison reverse enables split horizon on the interface.	page 849
Advertising and learning specific routes	You can control the routes that a device learns or advertises.	The device learns and advertises all RIP routes on all interfaces.	page 850

Configuring RIP parameters

Use the following procedures to configure RIP parameters on a system-wide and individual interface basis.

Enabling RIP

RIP is disabled by default. To enable RIP, you must enable it globally and also on individual interfaces on which you want to advertise RIP. Globally enabling the protocol does not enable it on individual interfaces. You can enable the protocol on physical interfaces as well as virtual routing interfaces. When you enable RIP on a port, you also must specify the version (version 1 only, version 2 only, or version 1 compatible with version 2).

To enable RIP globally, enter the following command.

```
NetIron(config)# router rip
```

Syntax: [no] router rip

After globally enabling the protocol, you must enable it on individual interfaces. To enable RIP on an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# ip rip v1-only
```

Syntax: [no] ip rip v1-only | v1-compatible-v2 | v2-only

Configuring metric parameters

By default, a PowerConnect port increases the cost of a RIP route that is learned or advertised on the port by one. You can configure individual ports to add more than one to a learned or advertised route's cost.

Changing the cost of routes learned or advertised on a port

By default, a PowerConnect port increases the cost of a RIP route that is learned on the port. The PowerConnect increases the cost by adding one to the route's metric before storing the route.

You can change the amount that an individual port adds to the metric of RIP routes learned on the port.

To increase the metric for learned routes, enter commands such as the following.

```
NetIron(config-if-e1000-1/1)# ip rip metric-offset 5 in
```

The command configures port 1/1 to add 5 to the cost of each route it learns.

Syntax: [no] ip rip metric-offset <*num*> **in | out**

The number is 1-16. A route with a metric of 16 is unreachable. Use 16 only if you do not want the route to be used. In fact, you can prevent the device from using a specific port for routes learned though that port by setting its metric to 16.

In applies to routes the port learns from RIP neighbors.

Out applies to routes the port advertises to its RIP neighbors.

Changing the administrative distance

By default, the device assigns the default RIP administrative distance (120) to RIP routes. When comparing routes based on administrative distance, the device selects the route with the lower distance. You can change the administrative distance for RIP routes.

NOTE
Refer to "Changing administrative distances" on page 1030 for a list of the default distances for all route sources.

To change the administrative distance for RIP routes, enter a command such as the following.

```
NetIron(config-rip-router)# distance 140
```

The command changes the administrative distance to 140 for all RIP routes.

Syntax: [no] distance <*number*>

The number is 1 - 255.

Configuring redistribution

You can configure the device to redistribute routes learned through OSPF or BGP4, connected into RIP, or static routes. When you redistribute a route from one of these other protocols into RIP, the device can use RIP to advertise the route to its RIP neighbors.

To configure redistribution, perform the following tasks:

- Configure redistribution filters. You can configure filters to permit or deny redistribution for a route based on its origin (OSPF, BGP4, and so on), the destination network address, and the route's metric. You also can configure a filter to set the metric based on these criteria.
- Change the default redistribution metric (optional). The device assigns a RIP metric of one to each redistributed route by default. You can change the default metric to a value up to 16.

Configuring redistribution filters

RIP redistribution filters apply to all interfaces. You use route maps to define how you want to deny or permit redistribution.

NOTE
The default redistribution action is permit, even after you configure and apply redistribution filters to the virtual routing interface. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (the filter with the highest ID), then apply filters to allow specific routes.

A ***route map*** is a named set of match conditions and parameter settings that the router can use to modify route attributes and to control redistribution of the routes into other protocols. A route map consists of a sequence of up to 50 ***instances***. If you think of a route map as a table, an instance is a row in that table. The router evaluates a route according to a route map's instances in ascending numerical order. The route is first compared against instance 1, then against instance 2, and so on. As soon as a match is found, the router stops evaluating the route against the route map instances.

Route maps can contain ***match*** statements and ***set*** statements. Each route map contains a "permit" or "deny" action for routes that match the match statements:

- If the route map contains a permit action, a route that matches a match statement is permitted; otherwise, the route is denied.
- If the route map contains a deny action, a route that matches a match statement is denied.
- If a route does not match any match statements in the route map, the route is denied. This is the default action. To change the default action, configure the last match statement in the last instance of the route map to "permit any any".
- If there is no match statement, the software considers the route to be a match.
- For route maps that contain address filters, AS-path filters, or community filters, if the action specified by a filter conflicts with the action specified by the route map, the route map's action takes precedence over the individual filter's action.

If the route map contains set statements, routes that are permitted by the route map's match statements are modified according to the set statements.

In RIP, the match statements are based on prefix lists and access control lists. Set statements are based on tag values and metric values.

To configure redistribution filters, enter a command such as the following.

```
NetIron(config-rip-router)#redistribute bgp route-map longroute
```

Syntax: [no] redistribute connected | bgp | ospf | static [metric <*value*> | route-map <*name*>]

The **connected** parameter applies redistribution to connected types.

The **bgp** parameter applies redistribution to BGP4 routes.

The **ospf** parameter applies redistribution to OSPF routes.

The **static** parameter applies redistribution to IP static routes.

The **metric** <*value*> parameter sets the RIP metric value 1- 15 that will be applied to the routes imported into RIP.

The **route-map** <*name*> parameter indicates the route map's name.

Matching based on RIP protocol type

The **match** option has been added to the **route-map** command that allows statically configured routes or the routes learned from the IGP protocol RIP.

To configure the route map to match to RIP, enter a command such as the following.

```
NetIron(config-routemap test)# match protocol rip
```

Syntax: [no] match protocol rip

Changing the default redistribution metric

When the device redistributes a route into RIP, the software assigns a RIP metric (cost) to the route. By default, the software assigns a metric of one to each route that is redistributed into RIP. You can increase the metric that the device assigns, up to 15.

To change the RIP metric the device assigns to redistributed routes, enter a command such as the following.

```
NetIron(config-rip-router)# default-metric 10
```

This command assigns a RIP metric of 10 to each route that is redistributed into RIP.

Syntax: [no] default-metric <*1-15*>

Configuring route learning and advertising parameters

By default, a device learns routes from all its RIP neighbors and advertises RIP routes to those neighbors.

You can configure the following learning and advertising parameters:

- **Learning and advertising of RIP default routes** – The device learns and advertises RIP default routes by default. You can disable learning and advertising of default routes on a global or individual interface basis.
- **Learning of standard RIP routes** – By default, the device can learn RIP routes from all its RIP neighbors. You can configure RIP neighbor filters to explicitly permit or deny learning from specific neighbors.

Enabling learning of RIP default routes

By default, the device does not learn default RIP routes. You can enable learning of RIP default routes on a global or interface basis.

To enable learning of default RIP routes on a global basis, enter the following command.

```
NetIron(config-rip-router)# learn-default
```

Syntax: [no] learn-default

To enable learning of default RIP routes on an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# ip rip learn-default
```

Syntax: [no] ip rip learn-default

Configuring a RIP neighbor filter

By default, a device learns RIP routes from all its RIP neighbors. Neighbor filters allow you to specify the neighbor routers from which the device can receive RIP routes. Neighbor filters apply globally to all ports.

To configure a RIP neighbor filters, enter a command such as the following.

```
NetIron(config-rip-router)# neighbor 1 deny any
```

Syntax: [no] neighbor <*filter-num*> **permit | deny** <*source-ip-address*> **| any**

This command configures the NetIron so that the device does not learn any RIP routes from any RIP neighbors.

The following commands configure the device to learn routes from all neighbors except 192.168.1.170. Once you define a RIP neighbor filter, the default action changes from learning all routes from all neighbors to denying all routes from all neighbors except the ones you explicitly permit. Thus, to deny learning from a specific neighbor but allow all other neighbors, you must add a filter that allows learning from all neighbors. Make sure you add the filter to permit all neighbors as the last filter (the one with the highest filter number). Otherwise, the software can match on the permit all filter before a filter that denies a specific neighbor, and learn routes from that neighbor.

```
NetIron(config-rip-router)# neighbor 2 deny 192.16.1.170
NetIron(config-rip-router)# neighbor 1024 permit any
```

Changing the route loop prevention method

RIP uses the following methods to prevent routing loops:

- **Split horizon** – The device does not advertise a route on the same interface as the one on which the router learned the route. This is the default.
- **Poison reverse** – The device assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which the router learned the route.

These loop prevention methods are configurable on a global basis as well as on an individual interface basis. One of the methods is always in effect on an interface enabled for RIP. Thus, if you disable one method, the other method is enabled.

NOTE
These methods are in addition to RIP’s maximum valid route cost of 15.

To disable poison reverse and enable split horizon on a global basis, enter the following command.

```
NetIron(config-rip-router)# no poison-reverse
```

Syntax: [no] poison-reverse

To disable poison reverse and enable split horizon on an interface, enter commands such as the following.

```
NetIron(config-if-e10000-1/1)# no ip rip poison-reverse
```

Syntax: [no] ip rip poison-reverse

To disable split horizon and enable poison reverse on an interface, enter the command such as the following.

```
NetIron(config-if-e10000-1/1)# ip rip poison-reverse
```

You can configure the device to avoid routing loops by advertising local RIP routes with a cost of 16 ("infinite" or "unreachable") when these routes go down.

```
NetIron(config-rip-router)# poison-local-routes
```

Syntax: [no] poison-local-routes

Suppressing RIP route advertisement on a VRRP or VRRPE backup interface

NOTE
This section applies only if you configure the device for Virtual Router Redundancy Protocol (VRRP) or VRRP Extended (VRRPE). Refer to 17, “Configuring VRRP and VRRP-E”.

Normally, a VRRP or VRRPE Backup includes route information for the virtual IP address (the backed up interface) in RIP advertisements. As a result, other routers receive multiple paths for the backed up interface and might sometimes unsuccessfully use the path to the Backup rather than the path to the Master.

You can prevent the Backups from advertising route information for the backed up interface by enabling suppression of the advertisements.

To suppress RIP advertisements for the backed up interface in Router2, enter the following commands.

```
Router2(config)# router rip
Router2(config-rip-router)# use-vrrp-path
```

Syntax: [no] use-vrrp-path

The syntax is the same for VRRP and VRRPE.

Using prefix lists and route maps as route filters

You can configure prefix lists to permit or deny specific routes, then apply them globally or to individual interfaces and specify whether the lists apply to learned routes (in) or advertised routes (out).

You can configure route maps to permit or deny specific routes, then apply a route map to an interface, and specify whether the map applies to learned routes (in) or advertised routes (out).

NOTE
A route is defined by the destination's IP address and network mask.

NOTE
By default, routes that do not match a prefix list are learned or advertised. To prevent a route from being learned or advertised, you must configure a prefix list to deny the route.

To configure a prefix list, enter commands such as the following.

```
NetIron(config)# ip prefix-list list1  permit 192.53.4.1 255.255.255.0
NetIron(config)# ip prefix-list list2  permit 192.53.5.1 255.255.255.0
NetIron(config)# ip prefix-list list3  permit 192.53.6.1 255.255.255.0
NetIron(config)# ip prefix-list list4 deny 192.53.7.1 255.255.255.0
```

The prefix lists permit routes to three networks, and deny the route to one network.

Since the default action is permit, all other routes (routes not explicitly permitted or denied by the filters) can be learned or advertised.

Syntax: [no] ip prefix-list <*name*> **permit | deny** <*source-ip-address*> **| any** <*source-mask*> **| any**

To apply a prefix list at the global level of RIP, enter commands such as the following.

```
NetIron(config-rip-router)# prefix-list list1 in
```

Syntax: [no] prefix-list <*name*> **in | out**

To apply prefix lists to a RIP interface, enter commands such as the following.

```
NetIron(config-if-e1000-1/2)# ip rip prefix-list list2 in
NetIron(config-if-e1000-1/2)# ip rip prefix-list list3 out
```

Syntax: [no] ip rip prefix-list <*name*> **in | out**

In applies the prefix list to routes the device learns from its neighbor on the interface.

Out applies the prefix list to routes the device advertises to its neighbor on the interface.

The commands apply RIP list2 route filters to all routes learned from the RIP neighbor on port 1/2 and applies the lists to all routes advertised on port 1/2.

To apply a route map to a RIP interface, enter commands such as the following.

```
NetIron(config-if-e1000-1/2)# ip rip route-map map1 in
```

Syntax: [no] ip rip route-map <*name*> **in | out**

The **route-map** <*name*> can be a prefix list or an ACL. Setting this command can change the metric.

In applies the route map to routes the device learns from its neighbor on the interface.

Out applies the route map to routes the device advertises to its neighbor on the interface.

The commands apply route map map1 as route filters to routes learned from the RIP neighbor on port 1/2.

Setting RIP timers

You can set basic update timers for the RIP protocol. The protocol must be enabled in order to set the timers.

To set the timers.

```
NetIron(config) router rip
NetIron(config-rip-router)# timers 50
```

Syntax: [no] timers <*seconds*>

Possible values: 3 - 21845 seconds

Default: 30 seconds

The command specifies how often RIP update messages are sent.

Displaying RIP Information

To display RIP filters, enter the following command at any CLI level.

```
NetIron#> show ip rip
  RIP Summary
  Default port 520
      Administrative distance is 120
      Updates every 30 seconds, expire after 180
      Holddown lasts 180 seconds, garbage collect after 120
      Last broadcast 2, Next Update 26
      Need trigger update 0, Next trigger broadcast 3
      Minimum update interval 25, Max update Offset 5
      Split horizon is on; poison reverse is off
      Import metric 1
      Prefix List, Inbound : Not set
      Prefix List, Outbound : Not set
      Route-map, Inbound : Not set
      Route-map, Outbound : Not set
      Redistribute:
  No Neighbors are configured in RIP Neighbor Filter Table
```

Syntax: show ip rip

See Table 137 on page 853 for display information.

To display RIP filters for a specific interface, enter the following command.

```
NetIron#show ip rip interface
Interface eth 1/20
 Rip Mode : Version 2 Running: TRUE
 Route summarization disabled
 Split horizon is on; poison reverse is off
 Default routes not accepted
 Metric-offset, Inbound 1
 Metric-offset, Outbound 0
 Prefix List, Inbound : Not set
      Prefix List, Outbound : Not set
 Route-map, Inbound : Not set
Interface ve 10
 RIP Mode : Compatible Running: TRUE
 Route summarization disabled
 Split horizon is off; poison reverse is on
 Default routes not accepted
 Metric-offset, Inbound 1
 Metric-offset, Outbound 0
 Prefix List, Inbound : Not set
      Prefix List, Outbound : Not set
 Route-map, Inbound : Not set
 Route-map, Outbound : Not set
Interface ve 20
 RIP Mode : Version1 Running: TRUE
 Route summarization enabled
 Split horizon is off; poison reverse is on
 Default routes not accepted
 Metric-offset, Inbound 1
 Metric-offset, Outbound 0
 Prefix List, Inbound : Not set
      Prefix List, Outbound : Not set
 Route-map, Inbound : Not set
 Route-map, Outbound : Not set
```

Syntax: show ip rip interface <ifName>

This display shows the following information.

TABLE 137 CLI display of neighbor filter information

This field...	Displays...
RIP Summary area	Shows the current configuration of RIP on the device.
Statis metric	Shows the static metric configuration. ".not defined" means the route map has not been distributed.
OSPF metric	Shows what OSPF route map has been applied.
Neighbor Filter Table area	
Index	The filter number. You assign this number when you configure the filter.

TABLE 137 CLI display of neighbor filter information (Continued)

This field...	Displays...
Action	The action the router takes for RIP route packets to or from the specified neighbor: • deny – If the filter is applied to an interface's outbound filter group, the filter prevents the router from advertising RIP routes to the specified neighbor on that interface. If the filter is applied to an interface's inbound filter group, the filter prevents the router from receiving RIP updates from the specified neighbor. • permit – If the filter is applied to an interface's outbound filter group, the filter allows the router to advertise RIP routes to the specified neighbor on that interface. If the filter is applied to an interface's inbound filter group, the filter allows the router to receive RIP updates from the specified neighbor.
Neighbor IP Address	The IP address of the RIP neighbor.

To display RIP route information, enter the following command.

```
PowerConnect#show ip rip route
RIP Routing Table - 35 entries:
 1.0.0.0/8, from 51.1.0.2, ve 10 (2)
          RIP, metric 4, tag 0, timers: aging 17 holddown -163
 12.0.0.0/8, from 51.1.0.2, ve 10 (6)
          RIP, metric 16, tag 0, timers: holddown 19 garbage 19
 12.1.1.0/24, from 0.0.0.0, eth 1/20 (34)
          MCAST, metric 1, tag 0, timers: none
 51.1.0.0/24, from 0.0.0.0, ve 10 (1)
          MCAST, metric 1, tag 0, timers: none
```

Syntax: show ip rip route

To display current running configuration for interface 1/20, enter the following command.

```
NetIron#show running-config interface ethernet 1/20
interface ethernet 1/20
 enable
 ip ospf area 0
 ip ospf priority 0
 ip rip v2-only
 ip address 12.1.1.2/24
 ipv6 address 2000::1/32
 ipv6 enable
!
```

To display current running configuration for ve 10, enter the following command.

```
NetIron#show running-config interface ve 10
interface ve 10
 bfd interval 50 min-rx 50 multiplier 3
 ip ospf area 2
 ip rip v1-compatible-v2
 ip rip poison-reverse
 ip address 51.1.0.1/24
 ipv6 address 51:1:1::14/64
!
```

To display current running configuration for ve 20, enter the following command.

```
NetIron#show running-config interface ve 20
interface ve 20
 ip ospf area 1
 ip rip v1-only
 ip rip poison-reverse
 ip address 51.2.0.1/24
!
```

Configuring OSPF Version 2 24

Overview

The following list displays the OSPF features supported by PowerConnect B-MLXe:

- OSPF
- OSPF Graceful Restart
- OSPF Graceful Restart helper-mode
- OSPF Dynamic Metric Calculation for LAGs/VE
- OSPF Point-to-Point Links
- OSPF Non-Broadcast
- Router LSAs (Type 1)
- Network LSAs (Type 2)
- Interarea prefix LSAs for ABRs (Type 3)
- Interarearouter LSAs for ASBRs (Type 4)
- Autonomous system external LSAs (Type 5)
- Link LSAs (Type 8)
- Intra-area prefix LSAs (Type 9)
- OSPF Distribute List
- OSPF Administrative Distance Control Using Route Maps
- OSPF Non-stop routing (NSR)
- OSPFv3 Virtual-Link Enhancement:Dynamic Tunnel Calculation
- New encryption code for passwords, authentication keys, and community strings
- Support for the **show ip ospf interface** command with interface filters
- OSPF VRF-Lite for CE routers

OSPF is a link-state routing protocol. The protocol uses link-state advertisements (LSA) to update neighboring routers regarding its interfaces and information on those interfaces. The router floods these LSAs to all neighboring routers to update them regarding the interfaces. Each router maintains an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router.

The PowerConnect supports the following types of LSAs, which are described in RFC 2328 and 3101:

- Router link
- Network link
- Summary link
- Autonomous system (AS) summary link

- AS external link
- Not-So-Stubby Area (NSSA) external link

OSPF is built upon a hierarchy of network components. The highest level of the hierarchy is the ***Autonomous System (AS)***. An autonomous system is defined as a number of networks, all of which share the same routing and administration characteristics.

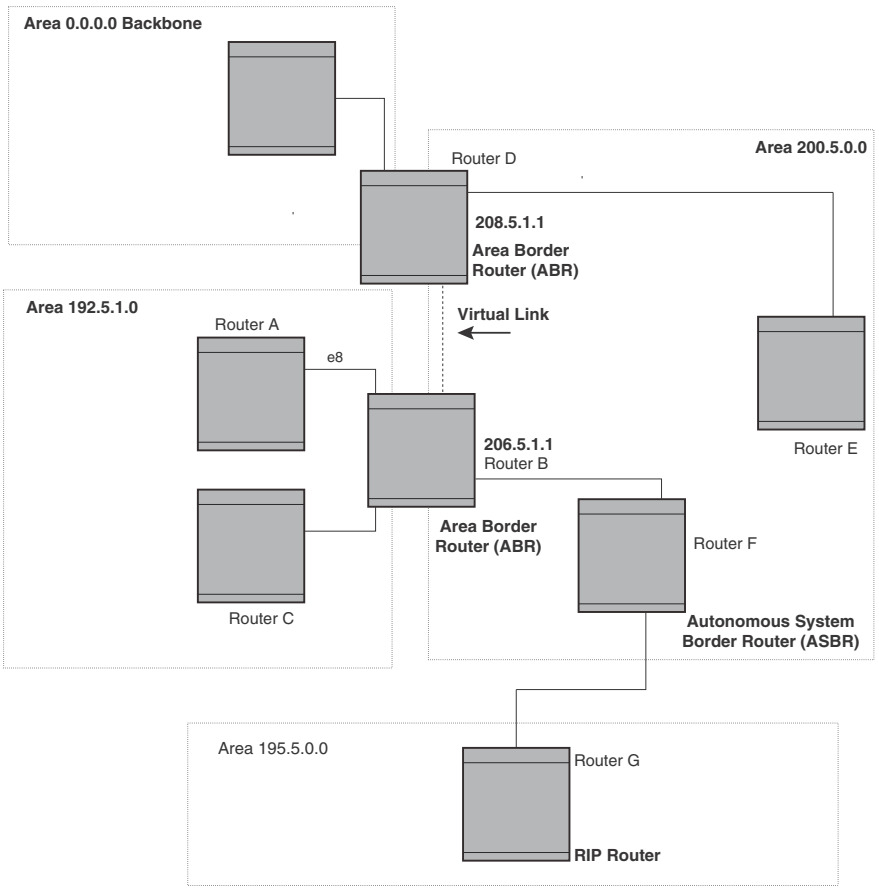
An AS can be divided into multiple ***areas*** as shown in Figure 132 on page 859. Each area represents a collection of contiguous networks and hosts. Areas limit the area to which link-state advertisements are broadcast, thereby limiting the amount of flooding that occurs within the network. An area is represented in OSPF by either an IP address or a number.

You can further limit the broadcast area of flooding by defining an area range. The area range allows you to assign an aggregate value to a range of IP addresses. This aggregate value becomes the address that is advertised instead all of the individual addresses it represents being advertised. You can assign up to 32 ranges in an OSPF area.

An OSPF router can be a member of multiple areas. Routers with membership in multiple areas are known as ***Area Border Routers (ABRs)***. Each ABR maintains a separate topological database for each area the router is in. Each topological database contains all of the LSA databases for each router within a given area. The routers within the same area have identical topological databases. The ABR is responsible for forwarding routing information or changes between its border areas.

An ***Autonomous System Boundary Router (ASBR)*** is a router that is running multiple protocols and serves as a gateway to routers outside an area and those operating with different protocols. The ASBR is able to import and translate different protocol routes into OSPF through a process known as ***redistribution***. For more details on redistribution and configuration examples, refer to "Enable route redistribution" on page 886

FIGURE 132 .OSPF operating in a network



OSPF point-to-point links

In an OSPF point-to-point network, where a direct Layer 3 connection exists between a single pair of OSPF routers, there is no need for Designated and Backup Designated Routers, as is the case in OSPF multi-access networks. Without the need for Designated and Backup Designated routers, a point-to-point network establishes adjacency and converges faster. The neighboring routers become adjacent whenever they can communicate directly. In contrast, in broadcast and non-broadcast multi-access (NBMA) networks, the Designated Router and Backup Designated Router become adjacent to all other routers attached to the network.

To configure an OSPF point-to-point link, refer to "Configuring an OSPF network type" on page 904.

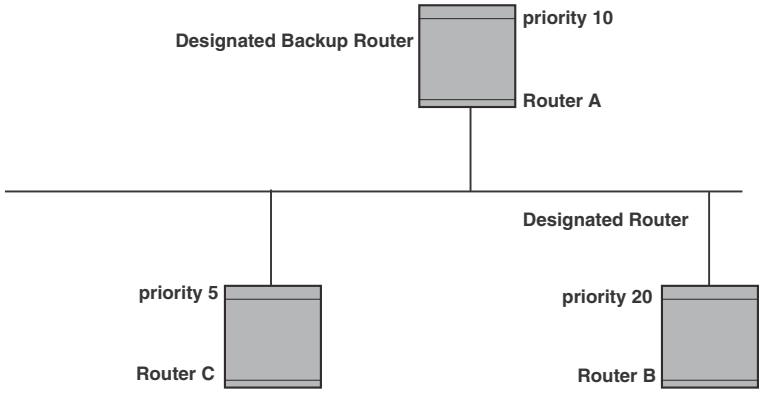
Designated routers in multi-access networks

In a network that has multiple routers attached, OSPF elects one router to serve as the designated router (DR) and another router on the segment to act as the backup designated router (BDR). This arrangement minimizes the amount of repetitive information that is forwarded on the network by forwarding all messages to the designated router and backup designated routers responsible for forwarding the updates throughout the network.

Designated router election in multi-access networks

In a network with no designated router and no backup designated router, the neighboring router with the highest priority is elected as the DR, and the router with the next largest priority is elected as the BDR, as shown in Figure 133

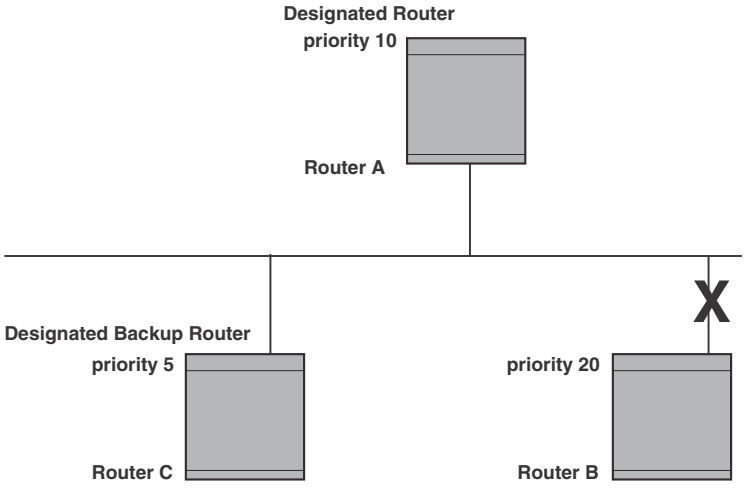
FIGURE 133 Designated and backup router election



If the DR goes off-line, the BDR automatically becomes the DR. The router with the next highest priority becomes the new BDR. This process is shown in Figure 134.

NOTE
Priority is a configurable option at the interface level. You can use this parameter to help bias one router as the DR.

FIGURE 134 Backup designated router becomes designated router



If two neighbors share the same priority, the router with the highest router ID is designated as the DR. The router with the next highest router ID is designated as the BDR.

NOTE
By default, the Dell router ID is the IP address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device.

When multiple routers on the same network are declaring themselves as DRs, then both priority and router ID are used to select the designated router and backup designated routers.

When only one router on the network claims the DR role despite neighboring routers with higher priorities or router IDs, this router remains the DR. This is also true for BDRs.

The DR and BDR election process is performed when one of the following events occurs:

- an interface is in a waiting state and the wait time expires
- an interface is in a waiting state and a hello packet is received that addresses the BDR
- a change in the neighbor state occurs, such as:
  - a neighbor state transitions from ATTEMPT state to a higher state
  - communication to a neighbor is lost
  - a neighbor declares itself to be the DR or BDR for the first time

OSPF RFC 1583 and 2328 compliance

Dell routers are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification. Dell routers can also be configured to operate with the latest OSPF standard, RFC 2328.

NOTE
For details on how to configure the system to operate with the RFC 2328, refer to "Modify OSPF standard compliance setting" on page 903.

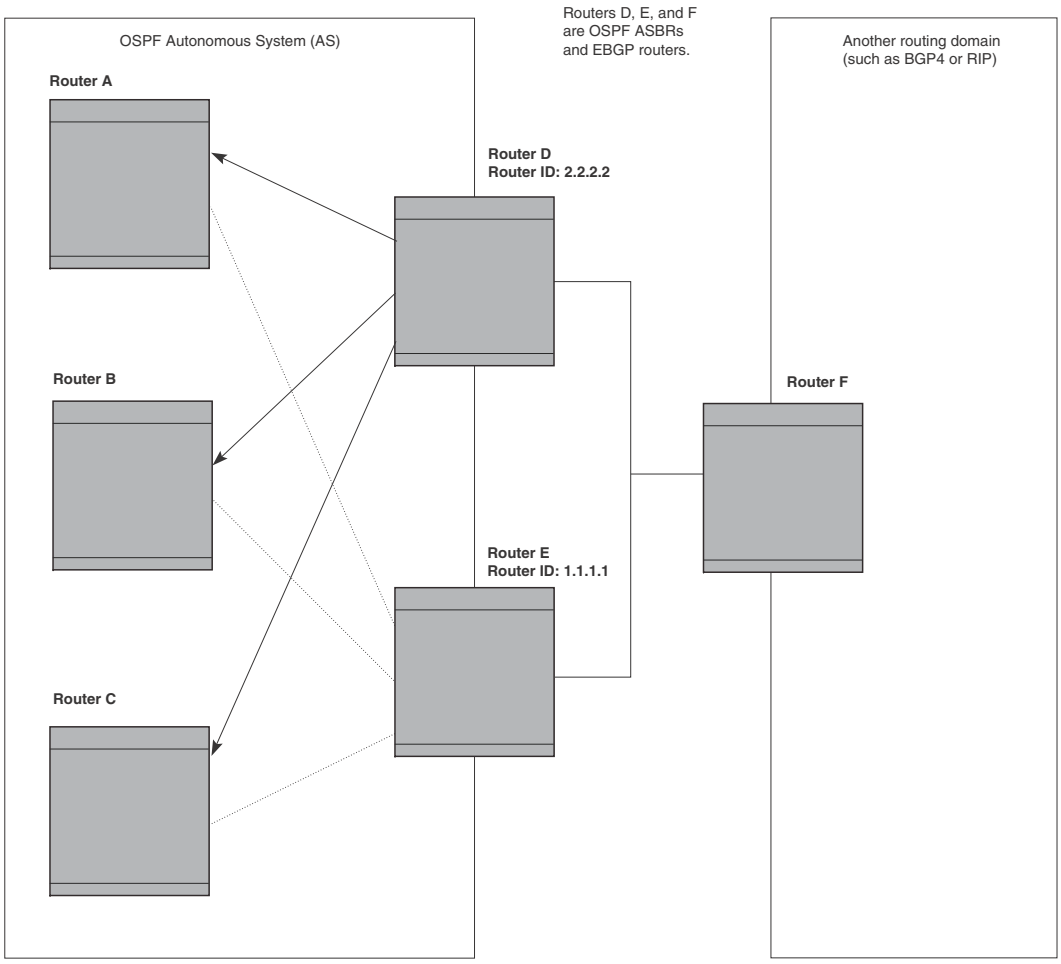
Reduction of equivalent AS external LSAs

An OSPF ASBR uses AS External link advertisements (AS External LSAs) to originate advertisements of a route learned from another routing domain, such as a BGP4 or RIP domain. The ASBR advertises the route to the external domain by flooding AS External LSAs to all the other OSPF routers (except those inside stub networks) within the local OSPF Autonomous System (AS).

In some cases, multiple ASBRs in an AS can originate equivalent LSAs. The LSAs are equivalent when they have the same cost, the same next hop, and the same destination. The device optimizes OSPF by eliminating duplicate AS External LSAs in this case. The device with the lower router ID flushes the duplicate External LSAs from its database and thus does not flood the duplicate External LSAs into the OSPF AS. AS External LSA reduction therefore reduces the size of the link state database on the device. The AS External LSA reduction is described in RFC 2328

Figure 135 shows an example of the AS External LSA reduction feature. In this example, Routers D and E are OSPF ASBRs, and thus communicate route information between the OSPF AS, which contains Routers A, B, and C, and another routing domain, which contains Router F. The other routing domain is running another routing protocol, such as BGP4 or RIP. Routers D, E, and F, therefore, are each running both OSPF and either BGP4 or RIP.

FIGURE 135 AS External LSA reduction



Notice that both Router D and Router E have a route to the other routing domain through Router F.

OSPF eliminates the duplicate AS External LSAs. When two or more devices are configured as ASBRs have equal-cost routes to the same next-hop router in an external routing domain, the ASBR with the highest router ID floods the AS External LSAs for the external domain into the OSPF AS, while the other ASBRs flush the equivalent AS External LSAs from their databases. As a result, the overall volume of route advertisement traffic within the AS is reduced and the devices that flush the duplicate AS External LSAs have more memory for other OSPF data. In Figure 135, since Router D has a higher router ID than Router E, Router D floods the AS External LSAs for Router F to Routers A, B, and C. Router E flushes the equivalent AS External LSAs from its database.

Algorithm for AS external LSA reduction

Figure 135 shows an example in which the normal AS External LSA reduction feature is in effect. The behavior changes under the following conditions:

- There is one ASBR advertising (originating) a route to the external destination, but one of the following happens:
  - A second ASBR comes on-line

- A second ASBR that is already on-line begins advertising an equivalent route to the same destination.

In either case above, the router with the higher router ID floods the AS External LSAs and the other router flushes its equivalent AS External LSAs. For example, if Router D is offline, Router E is the only source for a route to the external routing domain. When Router D comes on-line, it takes over flooding of the AS External LSAs to Router F, while Router E flushes its equivalent AS External LSAs to Router F.

- One of the ASBRs starts advertising a route that is no longer equivalent to the route the other ASBR is advertising. In this case, the ASBRs each flood AS External LSAs. Since the LSAs either no longer have the same cost or no longer have the same next-hop router, the LSAs are no longer equivalent, and the LSA reduction feature no longer applies.
- The ASBR with the higher router ID becomes unavailable or is reconfigured so that it is no longer an ASBR. In this case, the other ASBR floods the AS External LSAs. For example, if Router D goes off-line, then Router E starts flooding the AS with AS External LSAs for the route to Router F.

Support for OSPF RFC 2328 Appendix E

Dell devices support Appendix E in OSPF RFC 2328. Appendix E describes a method to ensure that an OSPF router generates unique link state IDs for type-5 (External) link state advertisements (LSAs) in cases where two networks have the same network address but different network masks.

NOTE
Support for Appendix E of RFC 2328 is enabled automatically and cannot be disabled. No user configuration is required.

Normally, an OSPF router uses the network address alone for the link state ID of the link state advertisement (LSA) for the network. For example, if the router needs to generate an LSA for network 10.1.2.3 255.0.0.0, the router generates ID 10.1.2.3 for the LSA.

However, suppose that an OSPF router needs to generate LSAs for all the following networks:

- 10.0.0.0 255.0.0.0
- 10.0.0.0 255.255.0.0
- 10.0.0.0 255.255.255.0

All three networks have the same network address, 10.0.0.0. Without support for RFC 2328 Appendix E, an OSPF router uses the same link state ID, 10.0.0.0, for the LSAs for all three networks. For example, if the router generates an LSA with ID 10.0.0.0 for network 10.0.0.0 255.0.0.0, this LSA conflicts with the LSA generated for network 10.0.0.0 255.255.0.0 or 10.0.0.0 255.255.255.0. The result is multiple LSAs that have the same ID but that contain different route information.

When appendix E is supported, the router generates the link state ID for a network as the following steps.

1. Does an LSA with the network address as its ID already exist?
   - No – Use the network address as the ID.
   - Yes – Go to step 2.

2. Compare the networks that have the same network address, to determine which network is more specific. The more specific network is the one that has more contiguous one bits in its network mask. For example, network 10.0.0.0 255.255.0.0 is more specific than network 10.0.0.0 255.0.0.0, because the first network has 16 ones bits (255.255.0.0) whereas the second network has only 8 ones bits (255.0.0.0).
   - For the less specific network, use the networks address as the ID.
   - For the more specific network, use the network's broadcast address as the ID. The broadcast address is the network address, with all ones bits in the host portion of the address. For example, the broadcast address for network 10.0.0.0 255.255.0.0 is 10.0.255.255.

   If this comparison results in a change to the ID of an LSA that has already been generated, the router generates a new LSA to replace the previous one. For example, if the router has already generated an LSA for network with ID 10.0.0.0 for network 10.0.0.0 255.255.255.0, the router must generate a new LSA for the network, if the router needs to generate an LSA for network 10.0.0.0 255.255.0.0 or 10.0.0.0 255.0.0.0.

OSPF graceful restart

The OSPF Graceful Restart feature provides support for high-availability routing. With this feature enabled, disruptions in forwarding are minimized and route flapping diminished to provide continuous service during times when a router experiences a restart.

With OSPF graceful restart enabled, a restarting router sends special LSAs to its neighbors called grace LSAs. These LSAs are sent to neighbors either before a planned OSPF restart or immediately after an unplanned restart. The grace LSA specifies a grace period for the neighbors of the restarting router to continue using the existing routes to and through the router after a restart. The restarting router comes up, it continues to use its existing OSPF routes as if nothing has occurred. In the background, the router re-acquires its neighbors prior to the restart and recalculates its OSPF routes and replaces them with new routes as necessary. Once the grace period has passed, the adjacent routers return to normal operation.

OSPF Graceful Restart can be enabled in the following configurations:

- **Configuring OSPF Graceful Restart for the Global Instance** – In this configuration all OSPF neighbors other than those used by VRFs are made subject to the Graceful Restart capability. The restart timer set globally does not apply to Graceful Restart on a configured VRF.
- **Configuring OSPF Graceful Restart per VRF** – In this configuration all OSPF neighbors for the specified VRF are made subject to the Graceful Restart capability. The restart timer set for a specific VRF only applies to that VRF.

Hitless upgrade support for OSPF graceful restart

OSPF graceful restart experiences minimal packet loss during hitless upgrade on a non-default VRF. On a default VRF, there is no packet loss during hitless upgrade.

OSPF Stub Router Advertisement

OSPFv2 Stub Router Advertisement is an open standard based feature and it is specified in RFC 3137. This feature provides a user with the ability to gracefully introduce and remove an OSPFv2 router from the network by controlling when the data traffic can start and stop flowing through the router in case where there are other OSPFv2 routers present on the network providing alternative paths for the traffic. This feature does not work if there is no alternative for the traffic through other OSPFv2 routers. The router can control the data traffic flowing through it by changing the cost of the paths passing through the configured router. By setting the path cost high the traffic will be redirected to other OSPFv2 routers providing a lower cost path. This change in path cost is accomplished by setting the metric of the links advertised in the Router LSA to a maximum value. When the OSPFv2 router is ready to forward the traffic the links are advertised with the real metric value instead of the maximum value.

The feature is useful for avoiding a loss of traffic during short periods when adjacency failures are detected and traffic is rerouted. Using this feature, traffic can be rerouted before an adjacency failure occurs due to common services interruptions such as a router being shutdown for maintenance.

The feature is also useful during router startup because it gives the router enough time to build up its routing table before forwarding traffic. This can be useful where BGP is enabled on the router because it takes time for the BGP routing table to converge.

Multi-Service IronWare software introduced an enhancement that allows you to configure and set a metric value for the following LSA types:

- Summary (type 3 and type 4)
- External (type 5 and type 7)
- Opaque (type 10, TE link)

Configuration of this feature is described in "Configuring OSPF router advertisement" on page 907.

OSPF Shortest Path First throttling

Multi-Service IronWare software introduced rapid triggering of SPF calculations with exponential back-off to offer the advantages of rapid convergence without sacrificing stability. As the delay increases, multiple topology changes can occur within a single SPF. This dampens network activity due to frequent topology changes.

This scheduling method starts with an initial value after which a configured delay time is followed. If a topology change event occurs the SPF is schedule after the time specified by the initial value, the router starts a timer for the time period specified by a configured hold time value. If no topology events occur during this hold time, the router returns to using the initial delay time.

If a topology event occurs during the hold time period, the next hold time period is recalculated to a value that is double the initial value. If no topology events occur during this extended hold time, the router resets to its initial value. If an event occurs during this extended hold time, the next hold time is doubled again. The doubling occurs as long as topology events occur during the calculated hold times until a configured maximum delay time value is reached or no event occurs (which resets the router to the initial hold time). The maximum value is then held until the hold time expires without a topology change event occurring. At any time that a hold time expires without a topology change event occurring, the router reverts to the initial hold value and begins the process all over again.

For example if you set the initial delay timer to 100 milliseconds, the hold timer to 300 and the maximum hold timer to 2000 milliseconds, the following would occur:

If a topology change occurs the initial delay of 100 milliseconds will be observed. If a topology change occurs during the hold time of 300 milliseconds the hold time is doubled to 600 milliseconds. If a topology change event occurs during the 600 millisecond period, the hold time is doubled again to 1200 milliseconds. If a topology change event occurs during the 1200 millisecond period, the hold time is doubled to 2400 milliseconds. Because the maximum hold time is specified as 2000, the value will be held at 2000. This 2000 millisecond period will then repeat as long as topology events occur within the maximum 2000 millisecond hold time. When a maximum hold time expires without a topology event occurring, the router reverts to the initial delay time and the cycle repeats as described.

The purpose of this feature is to use longer SPF scheduling values during network topology instability.

Configuration of this feature is described in "Configuring OSPF shortest path first throttling" on page 909.

IETF RFC and internet draft support

The implementation of OSPF Graceful Restart supports the following IETF RFC:

- RFC 3623: Graceful OSPF Restart

NOTE
A secondary management module must be installed for the device to function as a graceful restart device. If the device functions as a graceful restart helper device only, there is no requirement for a secondary management module.

For details on how to configure OSPF Graceful Restart, refer to "Configuring OSPF Graceful Restart" on page 905.

Dynamic OSPF activation and configuration

OSPF is automatically activated when you enable it. The protocol does not require a software reload.

You can configure and save the following OSPF changes without resetting the system:

- All OSPF interface-related parameters (for example: area, hello timer, router dead time cost, priority, re-transmission time, transit delay)
- All area parameters
- All area range parameters
- All virtual-link parameters
- All global parameters
- creation and deletion of an area, interface or virtual link
- Changes to address ranges
- Changes to global values for redistribution
- Addition of new virtual links

OSPF VRF-Lite for customer-edge routers

When a type 3, 5, or 7 LSA is sent from a provider edge (PE) router to a customer edge (CE) router, the DN (down) bit in the LSA options field must be set. This prevents any type 3, 5, or 7 LSA messages sent from the CE router to the PE router from being distributed any farther. The PE router ignores messages with the DN bit set and does not add these routes to the VRF routing table.

When you enable VRF-Lite on the CE router, the DN setting is ignored, allowing the CE router to add these routes to the VRF routing table.

To enable VRF-Lite, enter commands such as the following:

```
NetIron(config)# router ospf vrf 1
NetIron(config-ospf-router-vrf-1)# vrf-lite-capability
```

Syntax: [no] vrf-lite-capability

Use the no form of the command to disable VRF-Lite. This applies to the VRF instance only. It does not apply to the default VRF.

NOTE
For vpn4 external routes to be installed on CE routers, the domain-tags on PE routers must be different that the domain-tags on CE routers.

NOTE
This command applies to CE routers only. This command does not apply to PE routers.

Configuring OSPF

To begin using OSPF on the router, perform the steps outlined below.

1. Enable OSPF on the router.
2. Assign the areas to which the router will be attached.
3. Assign individual interfaces to the OSPF areas.
4. Configure route map for route redistribution, if desired.
5. Enable redistribution, if desired.
6. Modify default global and port parameters as required.
7. Modify OSPF standard compliance, if desired.

Configuration rules

The configuration rules are as follows:

- If a router is to operate as an ASBR, you must enable the ASBR capability at the system level.
- Redistribution must be enabled on routers configured to operate as ASBRs.
- All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.

OSPF parameters

You can modify or set the following global and interface OSPF parameters.

Global parameters

The global OSPF parameters are as follows:

- Modify OSPF standard compliance setting.
- Assign an area.
- Define an area range.
- Define the area virtual link.
- Set global default metric for OSPF.
- Change the reference bandwidth for the default cost of OSPF interfaces.
- Disable or re-enable load sharing.
- Enable or disable default-information-originate.
- Modify Shortest Path First (SPF) timers
- Define external route summarization
- Define redistribution metric type.
- Define redistribution route maps.
- Enable redistribution.
- Change the LSA pacing interval.
- Modify OSPF Traps generated.
- Modify database overflow interval.
- Stub Router advertisement

Interface parameters

The interface OSPF parameters are as follows:

- Assign interfaces to an area.
- Define the authentication key for the interface.
- Change the authentication-change interval
- Modify the cost for a link.
- Modify the dead interval.
- Modify MD5 authentication key parameters.
- Modify the priority of the interface.
- Modify the retransmit interval for the interface.
- Modify the transit delay of the interface.

NOTE
You set global level parameters at the OSPF CONFIG Level of the CLI. To reach that level, enter **router ospf...** at the global CONFIG Level. Interface parameters for OSPF are set at the interface CONFIG Level using the CLI command, **ip ospf...**

Enable OSPF on the router

When you enable OSPF on the router, the protocol is automatically activated. To enable OSPF on the router, use the following method.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)#
```

This command launches you into the OSPF router level where you can assign areas and modify OSPF global parameters.

Note regarding disabling OSPF

If you disable OSPF, the device removes all the configuration information for the disabled protocol from the running configuration. Moreover, when you save the configuration to the startup configuration file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup configuration file.

The CLI displays a warning message such as the following.

```
NetIron(config-ospf-router)# no router ospf
router ospf mode now disabled. All ospf config data will be lost when writing to
flash!
```

If you have disabled the protocol but have not yet saved the configuration to the startup configuration file and reloaded the software, you can restore the configuration information by re-entering the **router ospf** command to enable the protocol. If you have already saved the configuration to the startup configuration file and reloaded the software, the information is gone.

If you are testing an OSPF configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup configuration file containing the protocol's configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup configuration file onto the flash memory.

Assign OSPF areas

Once OSPF is enabled on the system, you can assign areas. Assign an IP address or number as the ***area ID*** for each area. The area ID is representative of all IP addresses (subnets) on a router port. Each port on a router can support one area.

An area can be ***normal***, a ***stub***, or a ***Not-So-Stubby Area (NSSA)***:

- **Normal** – OSPF routers within a normal area can send and receive External Link State Advertisements (LSAs).
- **Stub** – OSPF routers within a stub area cannot send or receive External LSAs. In addition, OSPF routers in a stub area must use a default route to the area's Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) to send traffic out of the area.
- **NSSA** – The ASBR of an NSSA can import external route information into the area.
  - ASBRs redistribute (import) external routes into the NSSA as type 7 LSAs. Type-7 External LSAs are a special type of LSA generated only by ASBRs within an NSSA, and are flooded to all the routers within only that NSSA.
  - ABRs translate type 7 LSAs into type 5 External LSAs, which can then be flooded throughout the AS. You can configure address ranges on the ABR of an NSSA so that the ABR converts multiple type-7 External LSAs received from the NSSA into a single type-5 External LSA.

    When an NSSA contains more than one ABR, OSPF elects one of the ABRs to perform the LSA translation for NSSA. OSPF elects the ABR with the highest router ID. If the elected ABR becomes unavailable, OSPF automatically elects the ABR with the next highest router ID to take over translation of LSAs for the NSSA. The election process for NSSA ABRs is automatic.

Example

To set up the OSPF areas shown in Figure 132 on page 859, use the following method.

```
NetIron(config-ospf-router)# area 192.5.1.0
NetIron(config-ospf-router)# area 200.5.0.0
NetIron(config-ospf-router)# area 195.5.0.0
NetIron(config-ospf-router)# area 0.0.0.0
NetIron(config-ospf-router)# write memory
```

Syntax: [**no**] **area** <*num*> | <*ip-addr*>

The <*num*> | <*ip-addr*> parameters speciy the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 – 2,147,483,647.

Multi-Service IronWare software support up to 200 OSPF areas

Assign a totally stubby area

By default, the device sends summary LSAs (LSA type 3) into stub areas. You can further reduce the number of link state advertisements (LSA) sent into a stub area by configuring the device to stop sending summary LSAs (type 3 LSAs) into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs, but the device still accepts summary LSAs from OSPF neighbors and floods them to other neighbors. The device can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each router.

When you enter a command to disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the device flushes all of the summary LSAs it has generated (as an ABR) from the area.

NOTE
This feature applies only when the device is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

To disable summary LSAs for a stub area, enter commands such as the following.

```
NetIron(config-ospf-router)# area 40 stub 99 no-summary
```

Syntax: [no] area <*num*> **|** <*ip-addr*> **stub** <*cost*> **[no-summary]**

The <*num*> | <*ip-addr*> parameter specifies the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 – 2,147,483,647.

The **stub** <*cost*> parameter specifies an additional cost for using a route to or from this area and can be from 1 – 16777215. There is no default. Normal areas do not use the cost parameter.

The **no-summary** parameter applies only to stub areas and disables summary LSAs from being sent into the area.

Assign a Not-So-Stubby Area (NSSA)

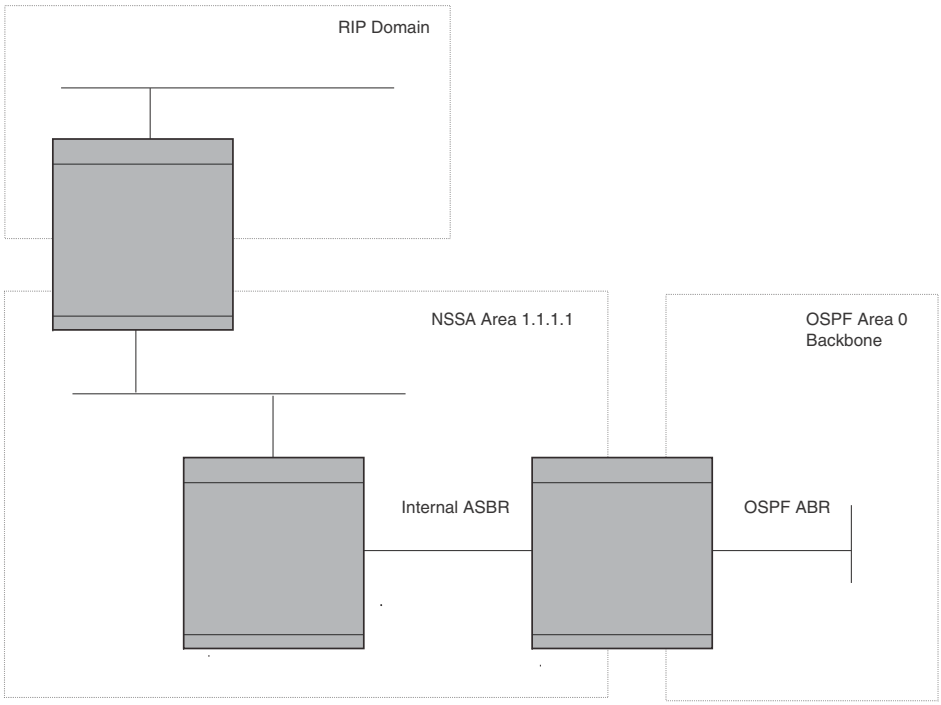
The OSPF Not So Stubby Area (NSSA) feature enables you to configure OSPF areas that provide the benefits of stub areas, but that also are capable of importing external route information. OSPF does not flood external routes from other areas into an NSSA, but does translate and flood route information from the NSSA into other areas such as the backbone.

NSSAs are especially useful when you want to summarize Type-5 External LSAs (external routes) before forwarding them into an OSPF area. The OSPF specification (RFC 2328) prohibits summarization of Type-5 LSAs and requires OSPF to flood Type-5 LSAs throughout a routing domain. When you configure an NSSA, you can specify an address range for aggregating the external routes that the NSSA's ABR exports into other areas.

The implementation of NSSA is based on RFC 1587.

Figure 136 shows an example of an OSPF network containing an NSSA.

FIGURE 136 OSPF network containing an NSSA



This example shows two routing domains, a RIP domain and an OSPF domain. The ASBR inside the NSSA imports external routes from RIP into the NSSA as Type-7 LSAs, which the ASBR floods throughout the NSSA.

The ABR translates the Type-7 LSAs into Type-5 LSAs. If an area range is configured for the NSSA, the ABR also summarizes the LSAs into an aggregate LSA before flooding the Type-5 LSAs into the backbone.

Since the NSSA is partially "stubby" the ABR does not flood external LSAs from the backbone into the NSSA. To provide access to the rest of the Autonomous System (AS), the ABR generates a default Type-7 LSA into the NSSA.

Configuring an NSSA

To configure OSPF area 1.1.1.1 as an NSSA, enter the following commands.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 1.1.1.1 nssa 1
NetIron(config-ospf-router)# write memory
```

Syntax: [no] area <*num*> **|** <*ip-addr*> **nssa** <*cost*> **[no-summary] | default-information-originate**

The <*num*> | <*ip-addr*> parameter specifies the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 – 2,147,483,647.

The **nssa** <*cost*> | **default-information-originate** parameter specifies that this is a Not-So-Stubby-Area (NSSA). The <*cost*> specifies an additional cost for using a route to or from this NSSA and can be from 1 – 16777215. There is no default. Normal areas do not use the cost parameter. Alternatively, you can use the **default-information-originate** parameter causes the device to inject the default route into the NSSA.

Specifying the **no-summary** option directs the router to not import type 3 summary LSAs into the NSSA area. The default operation is to import summary LSAs into an NSSA area.

NOTE
The device does not inject the default route into an NSSA by default.

To configure additional parameters for OSPF interfaces in the NSSA, use the **ip ospf area...** command at the interface level of the CLI.

Disabling the router to perform translations for NSSA LSAs

This command allows you to disable the router to perform translations for NSSA LSAs. When this command is used, type 7 NSSA external LSAs are not translated into type 5 external LSAs. This command is useful when the router is an area border router with many NSSA areas, and does not need to export the NSSA external routes into the backbone.

The following command enables this feature.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# no nssa-translator
```

Syntax: [no] nssa-translator

Configuring an address range for the NSSA

If you want the ABR that connects the NSSA to other areas to summarize the routes in the NSSA before translating them into Type-5 LSAs and flooding them into the other areas, configure an address range. The ABR creates an aggregate value based on the address range. The aggregate value becomes the address that the ABR advertises instead of advertising the individual addresses represented by the aggregate. You can configure up to 32 ranges in an OSPF area.

To configure an address range in NSSA 1.1.1.1, enter the following commands. This example assumes that you have already configured NSSA 1.1.1.1.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 1.1.1.1 range 209.157.22.1 255.255.0.0
NetIron(config-ospf-router)# write memory
```

Syntax: [no] area <*num*> **|** <*ip-addr*> **range** <*ip-addr*> <*ip-mask*> **[advertise | not-advertise]**

The <*num*> | <*ip-addr*> parameter specifies the area number, which can be in IP address format. If you specify a number, the number can be from 0 – 2,147,483,647.

The **range** <*ip-addr*> parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The <*ip-mask*> parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 209.157 are summarized into a single route.

The **advertise | not-advertise** parameter specifies whether you want the device to send type 3 LSAs for the specified range in this area. The default is **advertise**.

Assigning an area range (optional)

You can assign a ***range*** for an area, but it is not required. Ranges allow a specific IP address and mask to represent a range of IP addresses within an area, so that only that reference range address is advertised to the network, instead of all the addresses within that range. Each area can have up to 32 range addresses.

Example

To define an area range for subnets on 193.45.5.1 and 193.45.6.2, enter the following command.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 192.45.5.1 range 193.45.0.0 255.255.0.0
NetIron(config-ospf-router)# area 193.45.6.2 range 193.45.0.0 255.255.0.0
```

Syntax: [no] area <*num*> | <*ip-addr*> range <*ip-addr*> <*ip-mask*>

The <*num*> | <*ip-addr*> parameter specifies the area number, which can be in IP address format.

The range <*ip-addr*> parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The <*ip-mask*> parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 193.45 are summarized into a single route.

Assigning interfaces to an area

Once you define OSPF areas, you can assign interfaces to the areas. All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.

To assign interface 1/8 of Router A to area 192.5.0.0 and then save the changes, enter the following commands.

```
RouterA(config)# interface e 1/8
RouterA(config-if-e10000-1/8)# ip ospf area 192.5.0.0
RouterA(config-if-e10000-1/8)# write memory
```

Modify interface defaults

OSPF has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

Port default values can be modified using the following CLI commands at the interface configuration level of the CLI:

- ip ospf area <*ip-addr*>
- ip ospf auth-change-wait-time <*secs*>
- ip ospf authentication-key <*string*>
- ip ospf cost <*num*>
- ip ospf database-filter all out
- ip ospf dead-interval <*value*>

- ip ospf hello-interval <*value*>
- ip ospf md5-authentication key-activation-wait-time <*num*> | key-id <*num*> key <*string*>
- ip ospf mtu-ignore
- ip ospf passive
- ip ospf priority <*value*>
- ip ospf retransmit-interval <*value*>
- ip ospf transmit-delay <*value*>

For a complete description of these parameters, see the summary of OSPF port parameters in the next section.

OSPF interface parameters

The following parameters apply to OSPF interfaces

area	Assigns an interface to a specific area. You can assign either an IP address or number to represent an OSPF Area ID. If you assign a number, it can be any value from 0 – 2,147,483,647.
auth-change-wait-time	OSPF gracefully implements authentication changes to allow all routers to implement the change and thus prevent disruption to neighbor adjacencies. During the authentication-change interval, both the old and new authentication information is supported. The default authentication-change interval is 300 seconds (5 minutes). You change the interval to a value from 0 – 14400 seconds.

authentication-key <*string*>	By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **key** and <*string*>. For example,
	`NetIron(config-if-e10000-1/8)# ip ospf authentication-key 0 morningadmin`
	The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code "2".
	`ip ospf authentication-key 2 $on-o`
	The prefix can be one of the following:
	• 0 = the key string is not encrypted and is in clear text
	• 1 = the key string uses proprietary simple crytographic 2-way algorithm
	• 2 = the key string uses proprietary base64 crytographic 2-way algorithm

cost	Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps, 1Gbps, and 10 Gbps. The default cost is calculated by dividing 100 million by the bandwidth. For 10 Mbps links, the cost is 10. The cost for 100 Mbps, 1Gbps, and 10 Gbps links is 1, because the speed of 100 Mbps and 10Gbps was not in use at the time the OSPF cost formula was devised.
database-filter	Blocks all outbound LSAs on the OSPF interface.
dead-interval:	Indicates the number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The value can be from 1 – 65535 seconds. The default is 40 seconds. The rules described in "Rules for OSPF dead interval and hello interval timers" on page 877 apply regarding this timer.

hello-interval	Represents the length of time between the transmission of hello packets. The value can be from 1 – 65535 seconds. The default is 10 seconds. The rules described in "Rules for OSPF dead interval and hello interval timers" on page 877 apply regarding this timer.
MD5-authentication activation wait time	The number of seconds the device waits until placing a new MD5 key into effect. The wait time provides a way to gracefully transition from one MD5 key to another without disturbing the network. The wait time can be from 0 – 14400 seconds. The default is 300 seconds (5 minutes).
MD5-authentication key <*string*>	The MD5 **key** is a number from 1 – 255 and identifies the MD5 key that is being used. This parameter is required to differentiate among multiple keys defined on a router. By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **key** and <*string*>. For example, NetIron`(config-if-e10000-1/8)# ip ospf 1 md-5-authentication key-id 5 key` **`2`** `morningadmin` The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code "2". `ip ospf 1 md-5-authentication key-id 5 key` **`2`** `$on-o` The prefix can be one of the following: • 0 = the key string is not encrypted and is in clear text • 1 = the key string uses proprietary simple crytographic 2-way algorithm. • 2 = the key string uses proprietary base64 crytographic 2-way algorithm
mtu-ignore	A database description packet is rejected if the interface MTU specified in the DBD packet is greater than the MTU of the interface shared between the neighbors. To disable the mismatch condition set "mtu-ignore". By default, the mismatch detection is enabled
passive	When you configure an OSPF interface to be passive, that interface does not send or receive OSPF route updates. By default, all OSPF interfaces are active and thus can send and receive OSPF route information. Since a passive interface does not send or receive route information, the interface is in effect a stub network. OSPF interfaces are active by default. **Note:** This option affects all IP subnets configured on the interface. If you want to disable OSPF updates only on some of the IP subnets on the interface, use the **ospf-ignore** or **ospf-passive** parameter with the **ip address** command.
priority	Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The value can be from 0 – 255. The default is 1. If you set the priority to 0, the device does not participate in DR and BDR election.
retransmit-interval	The time between retransmissions of link-state advertisements (LSAs) to adjacent routers for this interface. The value can be from 0 – 3600 seconds. The default is 5 seconds.
transit-delay	The time it takes to transmit Link State Update packets on this interface. The value can be from 0 – 3600 seconds. The default is 1 second.

Rules for OSPF dead interval and hello interval timers

The following rules apply regarding these timers:

- If both the **hello-interval** and **dead-interval** parameters are configured, they will each be set to the values that you have configured.
- If the **hello-interval** parameter is configured, but not the **dead-interval** parameter, the **dead-interval** parameter will be set to a value that is 4 times the value set for the **hello-interval.**
- If the **dead-interval** parameter is configured, but not the **hello-interval** parameter, the **hello-interval.** parameter will be set to a value that is 1/4 the value set for the **dead-interval.** The minimum value for the **hello-interval** is 1.

Change the timer for OSPF authentication changes

When you make an OSPF authentication change, the software uses the authentication-change timer to gracefully implement the change. The software implements the change in the following ways:

- **Outgoing OSPF packets** – After you make the change, the software continues to use the old authentication to send packets, during the remainder of the current authentication-change interval. After this, the software uses the new authentication for sending packets.
- **Inbound OSPF packets** – The software accepts packets containing the new authentication and continues to accept packets containing the older authentication for two authentication-change intervals. After the second interval ends, the software accepts packets only if they contain the new authentication key.

The default authentication-change interval is 300 seconds (5 minutes). You change the interval to a value from 0 – 14400 seconds.

OSPF provides graceful authentication change for all the following types of authentication changes in OSPF:

- Changing authentication methods from one of the following to another of the following:
  - Simple text password
  - MD5 authentication
  - No authentication
- Configuring a new simple text password or MD5 authentication key
- Changing an existing simple text password or MD5 authentication key

To change the authentication-change interval, enter a command such as the following at the interface configuration level of the CLI.

```
NetIron(config-if-e10000-2/5)# ip ospf auth-change-wait-time 400
```

Syntax: [no] ip ospf auth-change-wait-time <*secs*>

The <*secs*> parameter specifies the interval and can be from 0 – 14400 seconds. The default is 300 seconds (5 minutes).

NOTE
For backward compatibility, the **ip ospf md5-authentication key-activation-wait-time** <*seconds*> command is still supported.

Block flooding of outbound LSAs on specific OSPF interfaces

By default, the device floods all outbound LSAs on all the OSPF interfaces within an area. You can configure a filter to block outbound LSAs on an OSPF interface. This feature is particularly useful when you want to block LSAs from some, but not all, of the interfaces attached to the area.

This command blocks all outbound LSAs. The command has been enhanced to provide options for selective blocking of LSAs.

After you apply filters to block the outbound LSAs, the filtering occurs during the database synchronization and flooding. When a filtering configuration is changed on a interface, all adjacencies on the interface are set to the Extstart state to restart the database exchange process. In cases where an LSA has already been flooded on an interface prior to application of the LSA filter, the LSA will not be flushed out from the remote neighbors. In this situation the user must clear the link state database and the adjacencies on all remote neighbors to flush out the leaked LSAs or wait for the LSAs to be aged out.

If you remove the filters, the blocked LSAs are automatically re-flooded. You do not need to reset OSPF to re-flood the LSAs.

NOTE
You cannot block LSAs on virtual links, and LSA filtering is not supported on sham links.

To apply a filter to an OSPF interface to block flooding of outbound LSAs on the interface, enter the following command at the Interface configuration level for that interface.

```
NetIron(config-if-e10000-1/1)# ip ospf database-filter all out
```

The command in this example blocks all outbound LSAs on the OSPF interface configured on port 1/1.

Syntax: [no] ip ospf database-filter {all | all-external [allow-default | allow-default-and-type4] | all-summary-external [allow-default | allow-default-and-type4] out

The **all** parameter directs the router to block all outbound LSAs on the OSPF interface.

The **all-external** option directs the router to allow the following LSAs: Router, Network, Opq-Area-TE, Opq-Link-Graceful and Type-3 Summary while it blocks all Type-4 and Type-5 LSAs unless directed by one of the following keywords:

allow-default – allows only Type-5 default LSAs .

allow-default-and-type4 – allows Type-5 default LSAs and all Type 4 LSAs.

The **all-summary-external** option directs the router to allow the following LSAs: Router, Network, Opq-Area-TE and Opq-Link-Graceful while it blocks all Type-3, Type-4 and Type-5 LSAs unless directed by one of the following keywords:

allow-default – allows only Type-3 or Type-5 default LSAs .

allow-default-and-type4 – allows Type-3 or Type-5 default LSAs and all Type 4 LSAs.

All Type-7 LSAs are always filtered if the **ip ospf database-filter** command is enabled.

By default, OSPF LSA filtering is disabled on all interfaces.

To remove the filter, enter a command such as the following.

```
NetIron(config-if-e10000-1/1)# no ip ospf database-filter all out
```

Assign virtual links

All ABRs (area border routers) must have either a direct or indirect link to the OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to the area backbone, the ABR can configure a ***virtual link*** to another router within the same area, which has a physical connection to the area backbone.

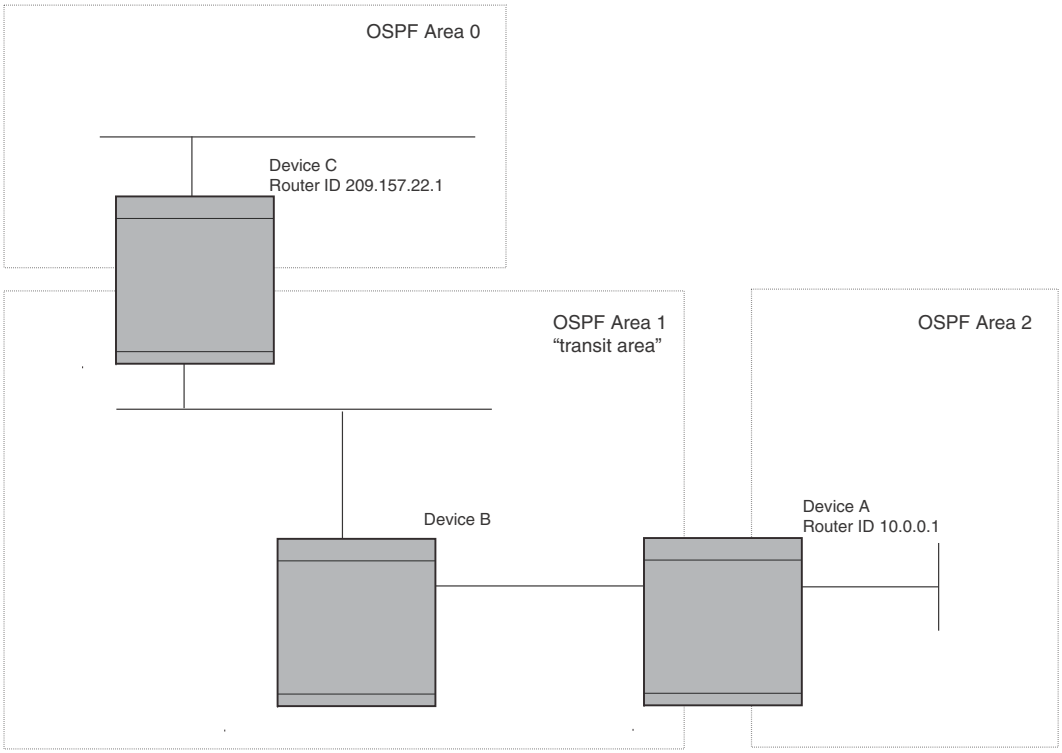
The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection), and the ABR requiring a logical connection to the backbone.

Two parameters fields must be defined for all virtual links—transit area ID and neighbor router:

- The ***transit area ID*** represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.
- The ***neighbor router*** field is the router ID (IP address) of the router that is physically connected to the backbone, when assigned from the router interface requiring a logical connection. When assigning the parameters from the router with the physical connection, the router ID is the IP address of the router requiring a logical connection to the backbone.

NOTE
By default, the Dell router ID is the IP address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

FIGURE 137 Defining OSPF virtual links within a network



Example

Figure 137 shows an OSPF area border router, Device A, that is cut off from the backbone area (area 0). To provide backbone access to Device A, you can add a virtual link between Device A and Device C using area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To define the virtual link on Device A, enter the following commands.

```
NetIron A(config)# router ospf
NetIron A(config-ospf-router)# area 2
NetIron A(config-ospf-router)# area 1
NetIron A(config-ospf-router)# area 1 virtual-link 209.157.22.1
NetIron A(config-ospf-router)# write memory
```

Enter the following commands to configure the virtual link on Device C.

```
NetIron C(config)# router ospf
NetIron C(config-ospf-router)# area 0
NetIron C(config-ospf-router)# area 1
NetIron C(config-ospf-router)# area 1 virtual-link 10.0.0.1
```

Syntax: [**no**] **area** <*ip-addr*> | <*num*> **virtual-link** <*router-id*>
[**authentication-key** <*string*> | **dead-interval** <*num*> | **hello-interval** <*num*> | **retransmit-interval** <*num*> | **transmit-delay** <*num*> | **md5-authentication key-activation-wait-time** <*num*> | **md5-authentication key-id** <*num*> **key** [**0**|**1**] <*string*>]

The **area** <*ip-addr*> | <*num*> parameters specify the transit area.

The **virtual-link** <*router-id*> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a device, enter the **show ip** command.

Refer to "Modify virtual link parameters" on page 881 for descriptions of the optional parameters.

Modify virtual link parameters

OSPF has some parameters that you can modify for virtual links. Notice that these are the same parameters as the ones you can modify for physical interfaces.

You can modify default values for virtual links using the following CLI command at the ***OSPF router level*** of the CLI, as shown in the following syntax:

Syntax: **[no] area** <*ip-addr*> **|** <*num*> **virtual-link** <*router-id*>
dead-interval <*num*> **| hello-interval** <*num*> **| retransmit-interval** <*num*> **|**
transmit-delay <*num*> **|**
authentication-key <*string*> **|**
md5-authentication key <*key-string*> **|**
md5-authentication key-activation-wait-time <*num*>

The parameters are described in the following table.

Virtual link parameter descriptions

You can modify the following virtual link interface parameters:

area <*ip-addr*> | <*num*>	The IP address or number of the transit area.
virutal-link <*router-id*>	The router ID of the OSPF router at the remote end of the virtual link.
dead-interval <*num*>	The number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The value can be from 1 – 65535 seconds. The default is 40 seconds. Refer to "Rules for OSPF dead interval and hello interval timers" on page 877 for more information about this timer.
hello-interval <*num*>	The length of time between the transmission of hello packets. The range is 1 – 65535 seconds. The default is 10 seconds.
retransmit-interval <*num*>	The interval between the re-transmission of link state advertisements to router adjacencies for this interface. The range is 0 – 3600 seconds. The default is 5 seconds.

transmit-delay <*num*>	The period of time it takes to transmit Link State Update packets on the interface. The range is 0 – 3600 seconds. The default is 1 second.

authentication-key <*string*>	This parameter allows you to assign different authentication encryption methods on a port-by-port basis. OSPF supports three methods of authentication for each interface: none, simple encryption, and base 64 encryption. Only one encryption method can be active on an interface at a time.
	The simple encryption and base 64 encryption methods requires you to configure an alphanumeric password on an interface. The password can be up to eight characters long. All OSPF packets transmitted on the interface contain this password. All OSPF packets received on the interface are checked for this password. If the password is not present, then the packet is dropped.
	By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **key** and <*string*>. For example,
	`NetIron C(config-ospf-router)# area 1 virtual-link 10.0.0.1 authentication-key 0 afternoon`
	The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code “2”.
	`area 1 virtual-link 12.12.12.25 authentication-key 2 $on-o`
	The prefix can be one of the following:
	• 0 = the key string is not encrypted and is in clear text
	• 1 = the key string uses proprietary simple crytographic 2-way algorithm
	• 2 = the key string uses proprietray base64 crytographic 2-way algorithm

md5-authentication key <*string*>	The MD5 **key** is a number from 1 – 255 and identifies the MD5 key that is being used. This parameter is required to differentiate among multiple keys defined on a router. When MD5 is enabled, the key-string is an alphanumeric password of up to 16 characters that is later encrypted and included in each OSPF packet transmitted. You must enter a password in this field when the system is configured to operate with either simple or MD5 authentication. By default, the MD5 authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **key** and <*string*>. For example, `NetIron C(config-ospf-router)# area 1 virtual-link 10.0.0.1 md-5-authentication key-id 5 key evening` The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code “2”. area 1 virtual-link 12.12.12.25 md-5-authentication key-id 5 key **2** $on-o The prefix can be one of the following: • 0 = the key string is not encrypted and is in clear text • 1 = the key string uses proprietary simple crytographic 2-way algorithm • 2 = the key string uses proprietary base64 crytographic 2-way algorithm
md5-authentication wait time	This parameter determines when a newly configured MD5 authentication key is valid. This parameter provides a graceful transition from one MD5 key to another without disturbing the network. All new packets transmitted after the key activation wait time interval use the newly configured MD5 Key. OSPF packets that contain the old MD5 key are accepted for up to five minutes after the new MD5 key is in operation. The range for the key activation wait time is from 0 – 14400 seconds. The default value is 300 seconds.

Changing the reference bandwidth for the cost on OSPF interfaces

Each interface on which OSPF is enabled has a cost associated with it. The device advertises its interfaces and their costs to OSPF neighbors. For example, if an interface has an OSPF cost of ten, the device advertises the interface with a cost of ten to other OSPF routers.

By default, an interface's OSPF cost is based on the port speed of the interface. The cost is calculated by dividing the reference bandwidth by the port speed. The default reference bandwidth is 100 Mbps, which results in the following default costs:

- 10 Mbps port – 10
- All other port speeds – 1

You can change the reference bandwidth, to change the costs calculated by the software.

The software uses the following formula to calculate the cost:

Cost = reference-bandwidth/interface-speed

If the resulting cost is less than 1, the software rounds the cost up to 1. The default reference bandwidth results in the following costs:

- 10 Mbps port's cost = 100/10 = 10
- 100 Mbps port's cost = 100/100 = 1
- 1000 Mbps port's cost = 100/1000 = 0.10, which is rounded up to 1
- 10 Gbps port's cost = 100/10000 = 0.01, which is rounded up to 1

The bandwidth for interfaces that consist of more than one physical port is calculated as follows:

- LAG group – The combined bandwidth of all the ports.
- Virtual interface – The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

The default reference bandwidth is 100 Mbps. You can change the reference bandwidth to a value from 1 – 4294967.

If a change to the reference bandwidth results in a cost change to an interface, the device sends a link-state update to update the costs of interfaces advertised by the device.

NOTE
If you specify the cost for an individual interface, the cost you specify overrides the cost calculated by the software.

Interface types to which the reference bandwidth does not apply

Some interface types are not affected by the reference bandwidth and always have the same cost regardless of the reference bandwidth in use:

- The cost of a loopback interface is always 1.
- The cost of a virtual link is calculated using the Shortest Path First (SPF) algorithm and is not affected by the auto-cost feature.
- The bandwidth for tunnel interfaces is 9 Kbps and is also subject to the auto-cost reference bandwidth setting.

Changing the reference bandwidth

To change the reference bandwidth, enter a command such as the following at the OSPF configuration level of the CLI.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# auto-cost reference-bandwidth 500
```

The reference bandwidth specified in this example results in the following costs:

- 10 Mbps port's cost = 500/10 = 50
- 100 Mbps port's cost = 500/100 = 5
- 1000 Mbps port's cost = 500/1000 = 0.5, which is rounded up to 1

The costs for 10 Mbps and 100 Mbps ports change as a result of the changed reference bandwidth. Costs for higher-speed interfaces remain the same.

Syntax: [no] auto-cost reference-bandwidth <*num*> **| use-active-ports**

The <*num*> parameter specifies the reference bandwidth and can be a value from 1 – 4294967. The default is 100.

To restore the reference bandwidth to its default value and thus restore the default costs of interfaces to their default values, enter the following command.

```
NetIron(config-ospf-router)# no auto-cost reference-bandwidth
```

Determining cost calculation for active ports only on LAG and VE interfaces

The default operation is for cost calculation of OSPF interfaces to be based upon all configured ports. There is also an option for the **auto-cost reference-bandwidth** command for the calculation of OSPF costs on active ports of LAG and VE interfaces. This option allows you to calculate cost based on the ports that are currently active. The following example enables cost calculation for currently active ports.

```
NetIron(config-ospf-router)# auto-cost use-active-ports
```

The **use-active-ports** option enables cost calculation for currently active ports only. This option does not have any effect on non-VE or non-LAG interfaces. The default operation is for costs to be based on configured ports.

Define redistribution filters

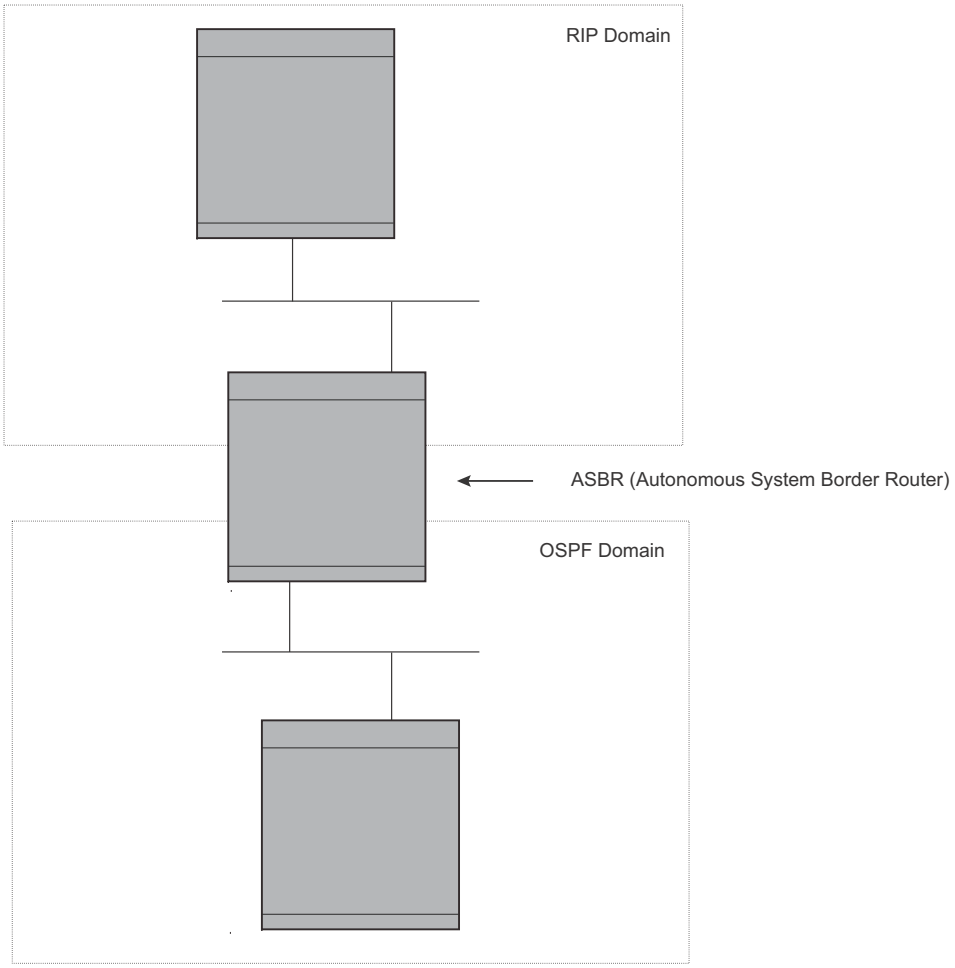
Route redistribution imports and translates different protocol routes into a specified protocol type. On the device, redistribution is supported for static routes, ISIS, OSPF, RIP, and BGP4. OSPF redistribution supports the import of static, ISIS, RIP, and BGP4 routes into OSPF routes.

NOTE
The device advertises the default route into OSPF even if redistribution is not enabled, and even if the default route is learned through an IBGP neighbor. IBGP routes (including the default route) are not redistributed into OSPF by OSPF redistribution (for example, by the OSPF **redistribute** command).

In Figure 138 on page 885, an administrator wants to configure the device acting as the ASBR (Autonomous System Boundary Router) between the RIP domain and the OSPF domain to redistribute routes between the two domains.

NOTE
The ASBR must be running both RIP and OSPF protocols to support this activity.

FIGURE 138 Redistributing OSPF and static routes to RIP routes



You also have the option of specifying import of just ISIS, RIP, OSPF, BGP4, or static routes, as well as specifying that only routes for a specific network or with a specific cost (metric) be imported, as shown in the command syntax below.

Syntax: [no] redistribute bgp | connected | rip | static [route-map <*map-name*>**]**

For example, to enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# redistribute rip
NetIron(config-ospf-router)# redistribute static
NetIron(config-ospf-router)# write memory
```

Modify default metric for redistribution

The default metric is a global parameter that specifies the cost applied to all OSPF routes by default. The default value is 10. You can assign a cost from 1 – 65535.

NOTE
You also can define the cost on individual interfaces. The interface cost overrides the default cost.

To assign a default metric of 4 to all routes imported into OSPF, enter the following commands.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# default-metric 4
```

Syntax: default-metric <*value*>

The <*value*> can be from 1 – 15. The default is 10.

Enable route redistribution

NOTE
Do not enable redistribution until you have configured the redistribution route map. Otherwise, you might accidentally overload the network with routes you did not intend to redistribute.

To enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# redistribute rip
NetIron(config-ospf-router)# redistribute static
NetIron(config-ospf-router)# write memory
```

Example using a route map

To configure a route map and use it for redistribution of routes into OSPF, enter commands such as the following.

```
NetIron(config)# ip route 1.1.0.0 255.255.0.0 207.95.7.30
NetIron(config)# ip route 1.2.0.0 255.255.0.0 207.95.7.30
NetIron(config)# ip route 1.3.0.0 255.255.0.0 207.95.7.30
NetIron(config)# ip route 4.1.0.0 255.255.0.0 207.95.6.30
NetIron(config)# ip route 4.2.0.0 255.255.0.0 207.95.6.30
NetIron(config)# ip route 4.3.0.0 255.255.0.0 207.95.6.30
NetIron(config)# ip route 4.4.0.0 255.255.0.0 207.95.6.30 5
NetIron(config)# route-map abc permit 1
NetIron(config-routemap abc)# match metric 5
NetIron(config-routemap abc)# set metric 8
NetIron(config-routemap abc)# router ospf
NetIron(config-ospf-router)# redistribute static route-map abc
```

The commands in this example configure some static IP routes, then configure a route map and use the route map for redistributing static IP routes into OSPF.

The **ip route** commands configure the static IP routes. The **route-map** command begins configuration of a route map called “abc”. The number indicates the route map entry (called the “instance”) you are configuring. A route map can contain multiple entries. The software compares routes to the route map entries in ascending numerical order and stops the comparison once a match is found.

The **match** command in the route map matches on routes that have 5 for their metric value (cost). The **set** command changes the metric in routes that match the route map to 8.

The **redistribute static** command enables redistribution of static IP routes into OSPF, and uses route map “abc“ to control the routes that are redistributed. In this example, the route map allows a static IP route to be redistributed into OSPF only if the route has a metric of 5, and changes the metric to 8 before placing the route into the OSPF route table.

The following command shows the result of the redistribution. Since only one of the static IP routes configured above matches the route map, only one route is redistributed. Notice that the route’s metric is 5 before redistribution but is 8 after redistribution.

```
NetIron# show ip ospf database external

Index Aging  LS ID           Router          Netmask  Metric    Flag
1     2      4.4.0.0         10.10.10.60     ffff0000 80000008  0000
```

Syntax: **[no] redistribute bgp | connected | rip | isis [level-1| level-1-2| level-2] | static [route-map** *<map-name>***]**

The **bgp | connected | rip | isis | static** parameter specifies the route source.

The **route-map** *<map-name>* parameter specifies the route map name. The following match parameters are valid for OSPF redistribution:

- **match ip address | next-hop** *<acl-num>*
- **match metric** *<num>*
- **match tag** *<tag-value>*

NOTE
A match tag can take up to 16 tags. During the execution of a route-map a match on any tag value in the list is considered a succesful match.

The following set parameters are valid for OSPF redistribution:

- **set ip next hop** *<ip-addr>*
- **set metric [+ | -]***<num>* **| none**
- **set metric-type type-1 | type-2**
- **set tag** *<tag-value>*

NOTE
You must configure the route map before you configure a redistribution that uses the route map.

NOTE
When you use a route map for route redistribution, the software disregards the permit or deny action of the route map.

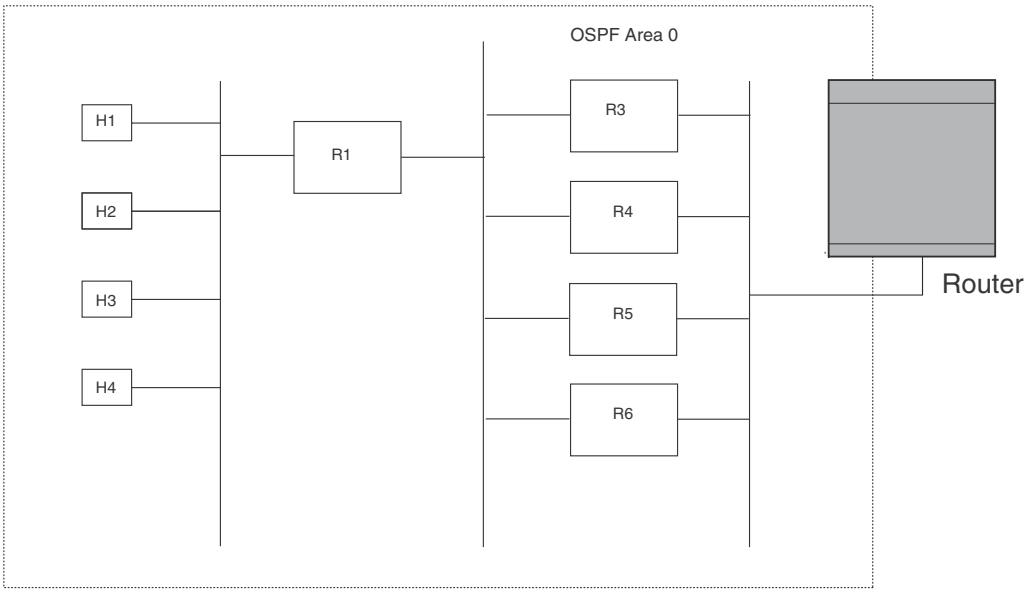
NOTE
For an external route that is redistributed into OSPF through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map. The **default-metric** <*num*> command has no effect on the route. This behavior is different from a route that is redistributed without using a route map. For a route redistributed without using a route map, the metric is set by the default-metric <*num*> command.

Disable or re-enable load sharing

Dell routers can load share among up to eight equal-cost IP routes to a destination. By default, IP load sharing is enabled. The default is 4 equal-cost paths but you can specify from 2 – 8 paths.

The router software can use the route information it learns through OSPF to determine the paths and costs. Figure 139 shows an example of an OSPF network containing multiple paths to a destination (in this case, R1).

FIGURE 139 Example OSPF network with four equal-cost paths



In the example in Figure 139, the device has four paths to R1:

- Router ->R3
- Router ->R4
- Router ->R5
- Router ->R6

Normally, the device will choose the path to the R1 with the lower metric. For example, if the metric for R3 is 1400 and the metric for R4 is 600, the device will always choose R4.

However, suppose the metric is the same for all four routers in this example. If the costs are the same, the router now has four equal-cost paths to R1. To allow the router to load share among the equal cost routes, enable IP load sharing. The software supports four equal-cost OSPF paths by default when you enable load sharing. You can specify from 2 – 8 paths.

NOTE
The device is not source routing in these examples. The device is concerned only with the paths to the next-hop routers, not the entire paths to the destination hosts.

OSPF load sharing is enabled by default when IP load sharing is enabled.

Configure external route summarization

When the device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to advertise one external route as an aggregate for all redistributed routes that are covered by a specified address range.

When you configure an address range, the range takes effect immediately. All the imported routes are summarized according to the configured address range. Imported routes that have already been advertised and that fall within the range are flushed out of the AS and a single route corresponding to the range is advertised.

If a route that falls within a configured address range is imported by the device, no action is taken if the device has already advertised the aggregate route; otherwise the device advertises the aggregate route. If an imported route that falls with in a configured address range is removed by the device, no action is taken if there are other imported routes that fall with in the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The device sets the forwarding address of the aggregate route to zero and sets the tag to zero.

If you delete an address range, the advertised aggregate route is flushed and all imported routes that fall within the range are advertised individually.

If an external LSDB overflow condition occurs, all aggregate routes are flushed out of the AS, along with other external routes. When the device exits the external LSDB overflow condition, all the imported routes are summarized according to the configured address ranges.

NOTE
If you use redistribution filters in addition to address ranges, the device applies the redistribution filters to routes first, then applies them to the address ranges.

NOTE
If you disable redistribution, all the aggregate routes are flushed, along with other imported routes.

NOTE
This option affects only imported, type 5 external routes. A single type 5 LSA is generated and flooded throughout the AS for multiple external routes. Type 7-route redistribution is not affected by this feature. All type 7 routes will be imported (if redistribution is enabled). To summarize type 7 LSAs or exported routes, use NSSA address range summarization.

To configure a summary address for OSPF routes, enter commands such as the following.

```
NetIron(config-ospf-router)# summary-address 10.1.0.0 255.255.0.0
```

The command in this example configures summary address 10.1.0.0, which includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. For all of these networks, only the address 10.1.0.0 is advertised in external LSAs.

Syntax: **summary-address** <*ip-addr*> <*ip-mask*>

The <*ip-addr*> parameter specifies the network address.

The <*ip-mask*> parameter specifies the network mask.

To display the configured summary addresses, enter the following command at any level of the CLI.

```
NetIron)# show ip ospf config
OSPF Redistribution Address Ranges currently defined:
Range-Address    Subnetmask
1.0.0.0          255.0.0.0
1.0.1.0          255.255.255.0
1.0.2.0          255.255.255.0
```

Syntax: show ip ospf config

Configure default route origination

When the device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to automatically generate a default external route into an OSPF routing domain. This feature is called "default route origination" or "default information origination".

By default, the device does not advertise the default route into the OSPF domain. If you want the device to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPF default route origination, the device advertises a type 5 default route that is flooded throughout the AS (except stub areas and NSSAs). In addition, internal NSSA ASBRs advertise their default routes as translatable type 7 default routes.

The device advertises the default route into OSPF even if OSPF route redistribution is not enabled, and even if the default route is learned through an IBGP neighbor.

NOTE
The device never advertises the OSPF default route, regardless of other configuration parameters, unless you explicitly enable default route origination using the following method.

If the device is an ASBR, you can use the "always" option when you enable the default route origination. The always option causes the ASBR to create and advertise a default route if it does not already have one configured.

If default route origination is enabled and you disable it, the default route originated by the device is flushed. Default routes generated by other OSPF routers are not affected. If you re-enable the feature, the feature takes effect immediately and thus does not require you to reload the software.

NOTE
The ABR (device) will not inject the default route into an NSSA by default and the command described in this section will not cause the device to inject the default route into the NSSA. To inject the default route into an NSSA, use the **area** <*num*> | <*ip-addr*> **nssa default-information-originate** command. Refer to "Assign a Not-So-Stubby Area (NSSA)" on page 871.

To enable default route origination, enter the following command.

```
NetIron(config-ospf-router)# default-information-originate
```

To disable the feature, enter the following command.

```
NetIron(config-ospf-router)# no default-information-originate
```

Syntax: [no] **default-information-originate** [**always**] [**metric** <*value*>] [**metric-type** <*type*>]

The **always** parameter advertises the default route regardless of whether the router has a default route. This option is disabled by default.

The **metric** <*value*> parameter specifies a metric for the default route. If this option is not used, the default metric is used for the route.

The **metric-type** <*type*> parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The <*type*> can be one of the following:

- type1 – Type 1 external route
- type2 – Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE
If you specify a metric and metric type, the values you specify are used even if you do not use the **always** option.

The **route-map** parameter overrides other options. If **set** commands for **metric** and **metric-type** are specified in the route-map, the command-line values of metric and metric-type if specified, are ignored" for clarification.

The **route-map** <rmap> parameter specifies the route map reference.

The corresponding route-map should be created before configuring the **route-map** option along with the **default-information-originate**. If the corresponding route-map was not been created beforehand, then the an error message will be displayed stating that the route-map must be created.

NOTE
The route-map option cannot be used with a non-default address in the match conditions. The default-route LSA shall not be generated if a default route is not present in the routing table and a **match ip address** condition for an existing non-default route is configured in the route-map. The **match ip-address** command in the route-map is a no-op operation for the default information originate command.

Supported match and set conditions

Table 138 and Table 139 list the supported **match** and **set** conditions of a normal route-map configuration:

TABLE 138 Match Conditions

Match Conditions	
ip nexthop prefix-list	<prefixList>
ip nexthop	<accessList>
interface	<interfaceName>
metric	<metricValue>
tag	<routeTagValue>
protocol-type	<protocol route type and (or) sub-type value>
route-type	<route type (IS-IS sub-type values)>

TABLE 139 Set Conditions

Set Conditions:	
metric	<metricValue>
metric-type	<type1/type2>
tag	<routeTagValue>

OSPF non-stop routing

The graceful restart feature supported by open shortest path first (OSPF) maintains area topology and dataflow. Though the network requires neighboring routers to support graceful restart and perform hitless failover, the graceful restart feature may not be supported by all routers in the network. NSR does not require support from neighboring routers to perform hitless failover.

If the active management module fails, the standby management module takes over and maintains the current OSPF routes, link-state advertisements (LSAs), and neighbor adjacencies, so that there is no loss of existing traffic to the OSPF destination.

Synchronization of critical OSPF elements

All types of LSAs and the neighbor information are synchronized to the standby module using the NSR synchronization library and IPC mechanism to transmit and receive packets.

Link state database synchronization

When the active management module fails, the standby management module takes over from the active management module with the identical OSPF link state database it had before the failure to ensure non-stop routing. The next shortest path firtst (SPF) run after switchover yieds the same result in routes as the active module had before the failure and OSPF protocol requires that all routers in the network to have identical databases.

LSA delayed acknowledging

When an OSPF router receives LSAs from its neighbor, it acknowledges the LSAs. After the acknowledgement is received, the neighbor removes this router from its retransmission list and stops resending the LSAs.

In the case of NSR, the router fails after receiving the LSA from its neighbor and has acknowledged that neighbor upon receipt of an LSA, and the LSA synchronization to the standby module is completed. In this case, the standby module when taking over from the active module does not have that LSA in its database and the already acknowledged neighbor does not retransmit that LSA. For this reason, the NSR-capable router waits for LSA synchronization of the standby module to complete (Sync-Ack) and then acknowledges the neighbor that sent the LSA.

LSA synching and packing

When the LSA processing is completed on the active management module and the decision is made to install the LSA in its link state database (LSDB), OSPF synchronizes that LSA to the standby module. OSPF checks the current state of the database entry whether or not it is marked for deletion. After checking the database state, OSPF packs the LSA status and other necessary information needed for direct installation in the standby OSPF LSDB along with the LSA portion. When the LSA reaches the standby module, OSPF checks the database entry state in the buffer and takes appropriate action, such as adding, overwriting, updating, or deleting the LSA from the LSDB.

Neighbor router synchronization

When the neighbor router is added in the active management module, it is synchronized and added to the standby module. When the neighbor is deleted in the active module, it is synchronized to the standby and deleted in the standby. When the neighbor router state becomes 2WAY or FULL, the neighbor router is synchronized to the standby module.The following attributes of the neighbor router is synchronized to the standby module:

- Neighbor router id
- Neighbor router ip address
- Destination router or backup destination router information
- Neighbor state 2WAY or FULL
- MD5 information
- Neighbor priority

Limitations

- If a neighbor router is inactive for 30 seconds, and if the standby module takes over in another 10 seconds, the neighbor router cannot be dropped. The inactivity timer starts again and takes another 40 seconds to drop the neighbor router.
- In standby module, the valid neighbor states are LOADING, DOWN, 2WAY, and FULL. If the active management processor (MP) fails when the neighbor state is LOADING, the standby module cannot continue from LOADING, but the standby can continue from 2WAY and tries to establish adjacency between the neighboring routers.

Interface synchronization

Interface information is synchronized for interfaces such as PTPT, broadcast, and non-broadcast. Interface wait time is not synchronized to the standby module. If an interface waits for 30 seconds to determine the identity of designated router (DR) or backup designated router (BDR), and if the standby module takes over, the wait timer starts again and takes another 40 seconds for the interface state to change from waiting to BDR, DR, or DROther.

BFD with OSPF NSR

Bidirectional forwarding detection (BFD) supports MP switchover and all BFD sessions for OSPF with graceful OSPF NSR, which are in the up state after the switchover. The BFD sessions for OSPF that do not use OSPF NSR are cleared before the switchover and then re-established on the new active MP after the MP switchover.

In case the active MP learns an OSPF neighbor and then restarts before a new BFD session is established, the standby module will not have a BFD session for the new OSPF neighbor. To overcome this and to support OSPF NSR with BFD, the following functions are supported when the active MP restarts:

- During MP switchover, BFD checks whether OSPF NSR is enabled. If OSPF NSR is enabled, the existing BFD sessions for OSPF is maintained during the switchover.
- OSPF sets up or clears the BFD sessions after OSPF neighbor transition.
- After the switchover, BFD sessions correspond with the active OSPF neighbor.

Standby module operations

The standby management module with OSPF configuration performs the following functions.

Neighbor database

Neighbor information is updated in the standby module based on updates from the active module. Certain neighbor state and interface transitions are synchronized to the standby module. By default, the neighbor timers on the standby module are disabled.

LSA database

The standby module processes LSA synchronization events from the active module and unpacks the LSA synchronization information to directly install it in its LSDB as the LSA has already been processed on the active module. The information required to install all types of LSAs (and special LSAs such as Grace LSAs) is packed by OSPF on the active module in the synchronization buffer, so that you can directly install LSAs on the standby module without extra processing.

The standby module is not allowed to originate any LSAs of its own. This is to maintain all information consistently from the active module. The active module synchronizes self-originated LSAs to the standby module.

LSA aging is not applicable on the standby module. During synchronization from the active, the current LSA age is recorded and the new database timestamp is created on the standby to later derive the LSA age as needed.

When the active module sends the LSAs to the standby module, based on the message, the standby module deletes or updates its link state database with the latest information.

LSA acknowledging or flooding are not done on the standby module. When the LSA synchronization update arrives from the active module, it will be directly installed into the LSDB.

Enabling and disabling NSR

To enable NSR for OSPF, enter the following command.

```
NetIron(config)# router ospf
PowerConnect(config-ospf-router)# nonstop-routing
```

To disable NSR for OSPF, enter the following command.

```
PowerConnect(config)# router ospf
```

`PowerConnect` (config-ospf-router)# no nonstop-routing

Syntax: [no] nonstop-routing

If you enter the **graceful-restart** command when NSR is already enabled, the command is rejected with the following message: “Error - Please disable NSR before enabling Graceful Restart”.

Similarly, if you enter the **nonstop-routing** command when graceful restart is already enabled, the command is rejected and the following message is displayed: “Error - Please disable Graceful Restart before enabling NSR”.

Limitations of NSR

Following are the limitations of NSR:

- Configurations that occur before the switchover are lost due to the CLI synchronization.
- Sham links are not supported.
- GRE tunnels are not supported.
- Changes in the neighbor state or interface state before or during a switchover do not take effect.
- Traffic counters are not synchronized because the neighbor and LSA database counters are recalculated on the standby module during synchronization.
- LSA acknowledging is delayed because it has to wait until standby acknowledging occurs.
- Depending on the sequence of redistribution or new LSAs (from neighbors), the LSAs accepted within the limits of the database may change after switchover.
- In NSR hitless failover, after switchover, additional flooding-related protocol traffic is generated to the directly connected neighbors.
- OSPF startup timers, database overflow, and max-metric, are not applied during NSR switchover.

Adding additional parameters

Previously, to add new parmeters, the old configuration had to be undone and the newer configuration had to be recreated. In release 04.1.00 however, to add new parmeters, the existing configuration need not be undone or removed. Any successive configuration changes with new parameters is appended to the existing configuration. If the same parmeter is entered again with a different value, then the corresponding parmeter value is updated.

Example

```
NetIron(config-ospf-router)#default-information-originate route-map defaultToOspf
NetIron(config-ospf-router)#default-information-originate always
NetIron(config-ospf-router)#default-information-originate metric 200
```

In the above example, **default-information-originate**' is enabled with the **route-map** parameter for the first CLI and then the **always** and **metric** is appended to the existing configuration. The running configuration of the above three split commands would be as follows:

```
NetIron(config-ospf-router)#default-information-originate always metric 200
route-map defaultToOspf
```

Disabling configuration

To disable the **route-map** parameter from the configuration, enter the following command:

```
NetIron(config-ospf-router)# no default-information-originate route-map
defaultToOspf
```

The above CLI would retain the configuration with **default-information-originate** alone and **route-map** option would get reset or removed.

The following commands with any or all of the options will remove the options from the **default-information-originate** command if any of the options are configured:

```
NetIron(config-ospf-router)#no default-information-originate always
```

```
NetIron(config-ospf-router)#no default-information-originate always route-map
test
```

```
NetIron(config-ospf-router)#no default-information-originate always route-map
test metric 200
```

```
NetIron(config-ospf-router)#no default-information-originate always route-map
test metric 200 metric-type type1
```

In the following example, the parameters of the **default-information-originate** command are reset if they are configured and if none of the parmeters are configured then, these commands will have no effect.

To disable the origination of default route, issue the command with "**no**" option and without any other options. This would remove the configuration of the **default information origination** even if any of the above mentioned options are configured.

Syntax: [no] default-information-originate [always] [metric <metric value>] [metric-type <metric-type>] [route-map <rmap-name>]

The **always** parameter advertises the default route regardless of whether the router has a default route. This option is disabled by default.

The **metric** <*value*> parameter specifies a metric for the default route. If this option is not used, the default metric is used for the route.

The **metric-type** <*type*> parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The <*type*> can be one of the following:

- type1 – Type 1 external route
- type2 – Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE
If you specify a metric and metric type, the values you specify are used even if you do not use the **always** option.

The **route-map** parameter overrides other options. If **set** commands for **metric** and **metric-type** are specified in the route-map, the command-line values of metric and metric-type if specified, are ignored" for clarification.

The **route-map** <rmap> parameter specifies the route map reference.

The corresponding route-map should be created before configuring the **route-map** option along with the **default-information-originate**. If the corresponding route-map was not been created beforehand, then the an error message will be displayed stating that the route-map must be created.

OSPF distribute list

This feature of PowerConnect Multi-Service IronWare configures a distribution list to explicitly deny specific routes from being eligible for installation in the IP route table. By default, all OSPF routes in the OSPF route table are eligible for installation in the IP route table. This feature does not block receipt of LSAs for the denied routes. The Layer 3 Switch still receives the routes and installs them in the OSPF database. The feature only prevents the software from installing the denied OSPF routes into the IP route table.

The OSPF distribution list can be managed using ACLs or Route Maps to identify routes to be denied as described in the following sections:

- Configuring an OSPF Distribution List using ACLs
- Configuring an OSPF Distribution List using Route Maps

Configuring an OSPF distribution list using ACLs

To configure an OSPF distribution list using ACLs:

- Configure an ACL that identifies the routes you want to deny. Using a standard ACL lets you deny routes based on the destination network, but does not filter based on the network mask. To also filter based on the destination network's network mask, use an extended ACL.
- Configure an OSPF distribution list that uses the ACL as input

Examples

In the following example, the first three commands configure a standard ACL that denies routes to any 78.x.x.x destination network and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 78.x.x.x destination network from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

```
NetIron(config)# ip access-list standard no_ip
NetIron(config-std-nacl)# deny 78.0.0.0 0.255.255.255
NetIron(config-std-nacl)# permit any
NetIron(config)# router ospf
NetIron(config-ospf-router) # area 0
NetIron(config-ospf-router) # distribute-list no_ip in
```

In the following example, the first three commands configure an extended ACL that denies routes to any 172.31.39.x destination network and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 172.31.39.x destination network from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

```
NetIron(config)# ip access-list extended DenyNet39
NetIron(config-ext-nacl)# deny ip 172.31.39.0 0.0.0.255 any
NetIron(config-ext-nacl)# permit ip any any
NetIron(config)# router ospf
NetIron(config-ospf-router) # area 0
NetIron(config-ospf-router) # distribute-list DenyNet39 in
```

In the following example, the first command configures a numbered ACL that denies routes to any 172.31.39.x destination network and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 172.31.39.x destination network from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

```
NetIron(config)# ip access-list 100 deny ip 172.31.39.0 0.0.0.255 any
NetIron(config)# ip access-list 100 permit ip any any
NetIron(config)# router ospf
NetIron(config-ospf-router) # area 0
NetIron(config-ospf-router) # distribute-list 100 in
```

Syntax: [no] distribute-list <*acl-name*> **|** <*acl-number*> **in**

The **distribute-list** command is applied globally to all interfaces on the router where it is executed.

Configuring an OSPF distribution list using route maps

You can manage an OSPF Distribution List using route maps that apply match operations as defined by an ACL or an IP prefix list. Additionally, you can also use other options available within the route maps and ACLs to further control the contents of the routes that OSPF provides to the IP route table. This section describes an example where an OSPF distribute list uses a route map to specify an OSPF Admin Distance for routes identified by an IP Prefix list.

To configure an OSPF distribution list using route maps:

- Configure a route map that identifies the routes you want to manage.
- Optionally configure an OSPF Admin Distance to apply to the OSPF routes
- Configure an OSPF distribution list that uses the route map as input

Example

In the following example, the first two commands identify two routes using the **ip prefix-list test1** command. Next, a **route-map** is created that uses the **prefix-list test1** to identify the two routes and the **set distance** command to set the OSPF Admin Distance of those routes to 200. A **distribute-list** is then configured under the OSPF configuration that uses the **route map** titled "setdistance" as input.

```
NetIron(config)# ip prefix-list test1 seq 5 permit 100.100.1.0/24
NetIron(config)# ip prefix-list test1 seq 10 permit 100.100.2.0/24
NetIron(config)# route-map setdistance permit 1
NetIron(config-routemap setdistance)# match ip address prefix-list test1
NetIron(config-routemap setdistance)# set distance 200
NetIron(config-routemap setdistance)# exit
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 0
NetIron(config-ospf-router)# area 1
NetIron(config-ospf-router)# distribute-list route-map setdistance in
NetIron(config-ospf-router)# exit
```

Once this configuration is implemented, the routes identified by the **ip prefix-list** command and matched in the Route Map will have their OSPF Admin Distance set to 200. This is displayed in the output from the **show ip route** command, as shown in the following.

```
NetIron# show ip route
Total number of IP routes: 4
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
        Destination        Gateway         Port         Cost           Type
1       11.1.1.0/24        DIRECT          eth 1/1      0/0             D
2       100.100.1.0/24     11.1.1.1        eth 1/1      200/2           O
3       100.100.2.0/24     11.1.1.1        eth 1/1      200/10          O2
4       100.100.6.0/24     11.1.1.1        eth 1/1      110/2           O
```

Routes 2 and 3 demonstrate the actions of the example configuration as both display an OSPF Admin Distance value of 200. Note that the value is applied to both OSPF learned routes that match the route-map configuration: internal (route 2) and external (route 3). The other OSPF internal route (route 4) that does not match the route-map continues to have the default OSPF admin distance of 110.

The following is an example of the **distribute-list** command applied with route-map **setdistance** set as the input.

```
NetIron(config-ospf-router)# distribute-list route-map setdistance in
```

Syntax: [no] distribute-list route-map <*routemap-name*> **in**

The <*routemap-name*> variable specifies the name of the route map being used to define the OSPF Distribute List.

The **distribute-list** command is applied to all OSPF LSAs on the router where it is executed.

NOTE
A Route Map used with the **distribute-list** command can use either the **ip prefix-list** command (as shown in the example) or an ACL to define the routes . For information about creating Route Maps, refer to 22, "Policy-Based Routing".

The **set distance** command that is used in association with a Route Map configuration.

Modify SPF timers

The device uses the following timers when calculating the shortest path for OSPF routes:

- **SPF delay** – When the device receives a topology change, the software waits before it starts a Shortest Path First (SPF) calculation. By default, the software waits 0 (zero) seconds. You can configure the SPF delay to a value from 0 – 65535 seconds. If you set the SPF delay to 0 seconds, the software immediately begins the SPF calculation after receiving a topology change.
- **SPF hold time** – The device waits for a specific amount of time between consecutive SPF calculations. By default, the device waits zero seconds. You can configure the SPF hold time to a value from 0 – 65535 seconds. If you set the SPF hold time to 0 seconds, the software does not wait between consecutive SPF calculations.

You can set the delay and hold time to lower values to cause the device to change to alternate paths more quickly in the event of a route failure. Note that lower values require more CPU processing time.

You can change one or both of the timers.

To change the SPF delay and hold time, enter commands such as the following.

```
NetIron(config-ospf-router)# timers spf 10 20
```

The command in this example changes the SPF delay to 10 seconds and changes the SPF hold time to 20 seconds.

To set the timers back to their default values, enter a command such as the following.

```
NetIron(config-ospf-router)# no timers spf 10 20
```

Syntax: [no] timers spf <*delay*> <*hold-time*>

The <*delay*> parameter specifies the SPF delay.

The <*hold-time*> parameter specifies the SPF hold time.

NOTE
OSPF incrementally updates the OSPF routing table when new Type-3 or Type-4 Summary, Type-5 External, or Type-7 External NSSA LSAs are received.

Modify redistribution metric type

The redistribution metric type is used by default for all routes imported into OSPF unless you specify different metrics for individual routes using redistribution filters. Type 2 specifies a big metric (three bytes). Type 1 specifies a small metric (two bytes). The default value is type 2.

To modify the default value to type 1, enter the following command.

```
NetIron(config-ospf-router)# metric-type type1
```

Syntax: [no] metric-type type1 | type2

The default is **type2**.

Modify administrative distance

The device can learn about networks from various protocols, including Border Gateway Protocol version 4 (BGP4), RIP, ISIS, and OSPF. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. The default administrative distance for OSPF routes is 110.

The router selects one route over another based on the source of the route information. To do so, the router can use the administrative distances assigned to the sources. You can bias the decision the device makes by changing the default administrative distance for OSPF routes.

Configuring administrative distance based on route type

You can configure a unique administrative distance for each type of OSPF route. For example, you can use this feature to prefer a static route over an OSPF inter-area route but you also want to prefer OSPF intra-area routes to static routes.

The distance you specify influences the choice of routes when the device has multiple routes for the same network from different protocols. The device prefers the route with the lower administrative distance.

You can specify unique default administrative distances for the following route types:

- Intra-area routes
- Inter-area routes
- External routes

The default for all these OSPF route types is 110.

NOTE
This feature does not influence the choice of routes within OSPF. For example, an OSPF intra-area route is always preferred over an OSPF inter-area route, even if the intra-area route's distance is greater than the inter-area route's distance.

To change the default administrative distances for inter-area routes, intra-area routes, and external routes, enter the following command.

```
NetIron(config-ospf-router)# distance external 100
NetIron(config-ospf-router)# distance inter-area 90
NetIron(config-ospf-router)# distance intra-area 80
```

Syntax: [no] distance external | inter-area | intra-area <*distance*>

The **external | inter-area | intra-area** parameter specifies the route type for which you are changing the default administrative distance.

The <*distance*> parameter specifies the new distance for the specified route type. Unless you change the distance for one of the route types using commands such as those shown above, the default is 110.

To reset the administrative distance to its system default (110), enter a command such as the following.

```
NetIron(config-ospf-router)# no distance external 100
```

Configure OSPF group Link State Advertisement (LSA) pacing

The device paces LSA refreshes by delaying the refreshes for a specified time interval instead of performing a refresh each time an individual LSA refresh timer expires. The accumulated LSAs constitute a group, which the device refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the device refreshes an accumulated group of LSAs, is configurable to a range from 10 – 1800 seconds (30 minutes). The default is 240 seconds (four minutes). Thus, every four minutes, the device refreshes the group of accumulated LSAs and sends the group together in the same packets.

Usage guidelines

The pacing interval is inversely proportional to the number of LSAs the device is refreshing and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval enhances performance. If you have a very small database (40 – 100 LSAs), increasing the pacing interval to 10 – 20 minutes might enhance performance slightly.

Changing the LSA pacing interval

To change the LSA pacing interval, use the following CLI method.

To change the LSA pacing interval to two minutes (120 seconds), enter the following command.

```
NetIron(config-ospf-router)# timers lsa-group-pacing 120
```

Syntax: [no] timers lsa-group-pacing <*secs*>

The <*secs*> parameter specifies the number of seconds and can be from 10 – 1800 (30 minutes). The default is 240 seconds (four minutes).

To restore the pacing interval to its default value, enter the following command.

```
NetIron(config-ospf-router)# no timers lsa-group-pacing
```

Modify OSPF traps generated

OSPF traps as defined by RFC 1850 are supported on device.

You can disable all or specific OSPF trap generation by entering the following CLI command.

```
NetIron(config)# no snmp-server trap ospf
```

To later re-enable the trap feature, enter **snmp-server trap ospf**.

To disable a specific OSPF trap, enter the command as **no snmp-server trap ospf** <*ospf-trap*>.

These commands are at the OSPF router Level of the CLI.

Here is a summary of OSPF traps supported on device, their corresponding CLI commands, and their associated MIB objects from RFC 1850. The first list are traps enabled by default:

- **interface-state-change-trap** – [MIB object: OspfIfstateChange]
- **virtual-interface-state-change-trap** – [MIB object: OspfVirtIfStateChange
- **neighbor-state-change-trap** – [MIB object:ospfNbrStateChange]

- **virtual-neighbor-state-change-trap** – [MIB object: ospfVirtNbrStateChange]
- **interface-config-error-trap** – [MIB object: ospfIfConfigError]
- **virtual-interface-config-error-trap** – [MIB object: ospfVirtIfConfigError]
- **interface-authentication-failure-trap** – [MIB object: ospfIfAuthFailure]
- **virtual-interface-authentication-failure-trap** – [MIB object: ospfVirtIfAuthFailure]
- **interface-receive-bad-packet-trap** – [MIB object: ospfIfrxBadPacket]
- **virtual-interface-receive-bad-packet-trap** – [MIB object: ospfVirtIfRxBadPacket]

The following traps are disabled by default.

- **interface-retransmit-packet-trap** – [MIB object: ospfTxRetransmit]
- **virtual-interface-retransmit-packet-trap** – [MIB object: ospfVirtIfTxRetransmit]
- **originate-lsa-trap** – [MIB object: ospfOriginateLsa]
- **originate-maxage-lsa-trap** – [MIB object: ospfMaxAgeLsa]
- **link-state-database-overflow-trap** – [MIB object: ospfLsdbOverflow]
- **link-state-database-approaching-overflow-trap** – [MIB object: ospfLsdbApproachingOverflow

Example

To stop an OSPF trap from being collected, use the CLI command: **no trap** <*ospf-trap*>, at the Router OSPF level of the CLI. To disable reporting of the neighbor-state-change-trap, enter the following command.

```
NetIron(config-ospf-router)# no trap neighbor-state-change-trap
```

Example

To reinstate the trap, enter the following command.

```
NetIron(config-ospf-router)# trap neighbor-state-change-trap
```

Syntax: [no] trap <*ospf-trap*>

Modify OSPF standard compliance setting

The device is configured, by default, to be compliant with the RFC 1583 OSPF V2 specification.

To configure a router to operate with the latest OSPF standard, RFC 2328, enter the following commands.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# no rfc1583-compatibility
```

Syntax: [no] rfc1583-compatibility

Modify exit overflow interval

If a database overflow condition occurs on a router, the router eliminates the condition by removing entries that originated on the router. The exit overflow interval allows you to set how often a device checks to see if the overflow condition has been eliminated. The default value is 0. The range is 0 – 86400 seconds (24 hours). If the configured value of the database overflow interval is zero, then the router never leaves the database overflow condition.

To modify the exit overflow interval to 60 seconds, enter the following command.

```
NetIron(config-ospf-router)# database-overflow-interval 60
```

Syntax: [no] database-overflow-interval <*value*>

The <*value*> can be from 0 – 86400 seconds. The default is 0 seconds.

Specify types of OSPF Syslog messages to log

You can specify which kinds of OSPF-related Syslog messages are logged. By default, the only OSPF messages that are logged are those indicating possible system errors. If you want other kinds of OSPF messages to be logged, you can configure the device to log them.

For example, to specify that all OSPF-related Syslog messages be logged, enter the following commands.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# log all
```

Syntax: [no] log all | adjacency [dr-only] | bad_packet [checksum] | database | memory | retransmit

The **log** command has the following options:

The **all** option causes all OSPF-related Syslog messages to be logged. If you later disable this option with the **no log all** command, the OSPF logging options return to their default settings.

The **adjacency** option logs essential OSPF neighbor state changes, especially on error cases. This option is disabled by default. The **dr-only** sub-option only logs essential OSPF neighbor state changes where the interface state is designated router (DR).

NOTE
For interfaces where the designated router state is not applicable, such as point-to-point and virtual links, OSPF neighbor state changes will always be logged irrespective of the setting of the **dr-only** sub-option.

NOTE
A limitation with the **dr-only** sub-option is that when a DR/BDR election is underway, OSPF neighbor state changes pertaining to non-DR/BDR routers are not logged. Logging resumes once a DR is elected on that network.

The **bad_packet checksum** option logs all OSPF packets that have checksum errors. This option is enabled by default.

The **bad_packet** option logs all other bad OSPF packets. This option is disabled by default.

The **database** option logs OSPF LSA-related information. This option is disabled by default.

The **memory** option logs abnormal OSPF memory usage. This option is enabled by default.

The **retransmit** option logs OSPF retransmission activities. This option is disabled by default.

Configuring an OSPF network type

To configure an OSPF network, enter commands such as the following.

```
NetIron(config)# interface eth 1/5
NetIron(config-if-1/5)# ip ospf network point-to-point
```

This command configures an OSPF point-to-point link on Interface 5 in slot 1.

Syntax: [no] ip ospf network point-to-point | broadcast | non-broadcast

The **point-to-point** option configures the network type as a point to point connection. This is the default option for POS interfaces.

NOTE
Dell supports numbered point-to-point networks, meaning the OSPF router must have an IP interface address which uniquely identifies the router over the network. Dell does not support unnumbered point-to-point networks.

The **broadcast** option configures the network type as a broadcast connection. This is the default option for Ethernet, VE and Loopback interfaces.

The **non-broadcast** option configures the network type as a non-broadcast connection. This allows you to configure the interface to send OSPF traffic to its neighbor as unicast packets rather than multicast packets. This can be useful in situations where multicast traffic is not feasible (for example when a firewall does not allow multicast packets).

On a non-broadcast interface, the routers at either end of this interface must configure non-broadcast interface type and the neighbor IP address. There is no restriction on the number of routers sharing a non-broadcast interface (for example, through a hub/switch).

To configure an OSPF interface as a non-broadcast interface, you enable the feature on a physical interface or a VE, following the **ip ospf area** statement, and then specify the IP address of the neighbor in the OSPF configuration. The non-broadcast interface configuration must be done on the OSPF routers at either end of the link.

For example, the following commands configure VE 20 as a non-broadcast interface.

```
NetIron(config)# int ve 20
NetIron(config-vif-20)# ip address 1.1.20.4/24
NetIron(config-vif-20)# ip ospf area 0
NetIron(config-vif-20)# ip ospf network non-broadcast
```

The following commands specify 1.1.20.1 as an OSPF neighbor address. The address specified must be in the same sub-net as the non-broadcast interface.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# neighbor 1.1.20.1
```

For example, to configure the feature in a network with three routers connected by a hub or switch, each router must have the linking interface configured as a non-broadcast interface, and the two other routers must be specified as neighbors.

Configuring OSPF Graceful Restart

OSPF Graceful Restart can be enabled in the following configurations:

- **Configuring OSPF Graceful Restart for the Global Instance** – In this configuration all OSPF neighbors other than those used by VRFs are made subject to the Graceful Restart capability. The restart timer set globally does not apply to Graceful Restart on a configured VRF.
- **Configuring OSPF Graceful Restart per VRF** – In this configuration all OSPF neighbors for the specified VRF are made subject to the Graceful Restart capability. The restart timer set for a specific VRF only applies to that VRF.

Configuring OSPF Graceful Restart for the global instance

OSPF Graceful restart can be configured for the global instance or for a specified Virtual Routing and Forwarding (VRF) instance. Configuring OSPF Graceful restart for the global instance does not configure it for any VRFs. The following sections describe how to enable the OSPF graceful restart feature for the global instance on a device.

Use the following command to enable the graceful restart feature for the global instance on a device.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# graceful-restart
```

Syntax: [no] **graceful-restart**

Configuring OSPF Graceful Restart time for the global instance

Use the following command to specify the maximum amount of time advertised to a neighbor router to maintain routes from and forward traffic to a restarting router.

```
NetIron(config) router ospf
NetIron(config-ospf-router)# graceful-restart restart-time 120
```

Syntax: [no] **graceful-restart restart-time** <*seconds*>

The <*seconds*> variable sets the maximum restart wait time advertised to neighbors.

Possible values are 10 - 1800 seconds.

The default value is 120 seconds.

Disabling OSPF Graceful Restart helper mode for the global instance

By default, a router supports other restarting routers as a helper. You can prevent your router from participating in OSPF Graceful Restart by using the following command.

```
NetIron(config) router ospf
NetIron(config-ospf-router)# graceful-restart helper-disable
```

Syntax: [no] **graceful-restart helper-disable**

This command disables OSPF Graceful Restart helper mode.

The default behavior is to help the restarting neighbors.

Configuring OSPF Graceful Restart per VRF

The following sections describe how to enable the OSPF Graceful Restart feature on a specified VRF.

Use the following command to enable the graceful restart feature on a specified VRF.

```
NetIron(config)# router ospf vrf blue
NetIron(config-ospf-router)# graceful-restart
```

Syntax: [no] **graceful-restart**

Configuring OSPF Graceful Restart time per VRF

Use the following command to specify the maximum amount of time advertised to an OSPF neighbor router to maintain routes from and forward traffic to a restarting router.

```
NetIron(config) router ospf vrf blue
NetIron(config-ospf-router)# graceful-restart restart-time 120
```

Syntax: [no] **graceful-restart restart-time** <*seconds*>

The <*seconds*> variable sets the maximum restart wait time advertised to OSPF neighbors of the VRF.

Possible values are 10 - 1200 seconds.

The default value is 60 seconds.

Disabling OSPF Graceful Restart helper mode per VRF

You can prevent your router from participating in OSPF Graceful Restart with VRF neighbors by using the following command.

```
NetIron(config) router ospf vrf blue
NetIron(config-ospf-router)# graceful-restart helper-disable
```

Syntax: [no] **graceful-restart helper-disable**

This command disables OSPF Graceful Restart helper mode.

The default behavior is to help the restarting neighbors.

For information about how to display OSPF Graceful Restart Information, refer to "Displaying an OSPF Graceful Restart information" on page 930.

Configuring OSPF router advertisement

You can configure OSPF router advertisement in the **router ospf** mode or **router ospf vrf** mode as shown in the following examples.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# max-metric router-lsa all-vrfs on-startup 30 link
all
NetIron(config)# router ospf vrf blue
NetIron(config-ospf-router)# max-metric router-lsa on-startup 30 link all
```

Syntax: [no] **max-metric router-lsa** [**all-vrfs**] [**on-startup** { <*time*> | **wait-for-bgp**}] [**summary-lsa** <*metric-value*>] [**external-lsa** <*metric-value*>] [**te-lsa** <*metric-value*>] [**all-lsas**] [**link** {**ptp** | **stub** | **transit** | **all** }]

The **all-vrfs** parameter specifies that the command will be applied to all VRF instances of OSPFv2. Note: this command is supported only for VRFs that are already configured when the **max-metric router-lsa all-vrfs** command is issued. Any new OSPF instance configured after the **max-metric** configuration is completed requires that the **max-metric** command be configured again to take in the new OSPF instance.

The **on-startup** parameter specifies that the OSPF router advertisement be performed at the next system startup. This is an optional parameter.

When using the **on-startup** option you can set a <*time*> in seconds for which the specified links in Router LSA will be advertised with the metric set to a maximum value of 0xFFFF. Optional values for <*time*> are 5 to 86400 seconds. There is no default value for <*time*> .

The **wait-for-bgp** option for the **on-startup** parameter directs OSPF to wait for either 600 seconds or until BGP has finished route table convergence (whichever event happens first), before advertising the links with the normal metric.

Using the **link** parameter you can specify the type of links for which the maximum metric is to be advertised. The default value is for maximum metric to be advertised for transit links only. This is an optional parameter.

With release 03.5.00 of the Multi-Service IronWare, additional options are supported that allow you to select the following LSA types and set the required metric:

The **summary-lsa** option specifies that the metric for all summary type 3 and type 4 LSAs will be modified to the specified <*metric-value*> or the default value. The range of possible values for the <*metric-value*> variable are 1 to 16777214 (Hex: 0x00001 to 0x00FFFFFE). The default value is 16711680 (Hex: 0x00FF0000).

The **external-lsa** option specifies that the metric for all external type 5 and type 7 LSAs will be modified to the specified <*metric-value*> or a default value. The range of possible values for the <*metric-value*> variable are 1 to 16777214 (Hex: 0x00001 to 0x00FFFFFE). The default value is 16711680 (Hex: 0x00FF0000).

The **te-lsa** option specifies that the TE metric field in the TE metric sub tlv for all type 10 Opaque LSAs LINK TLV originated by the router will be modified to the specified <*metric-value*> or a default value. The range of possible values for the <*metric-value*> variable are 1 to 4294967295 (Hex: 0x00001 to 0xFFFFFFFF). The default value is 4294967295 (Hex: 0xFFFFFFFF). This parameter only applies to the default instance of OSPF.

Examples

The following examples of the command max-metric router-lsa command demonstrate how it can be used:

The following command indicates that OSPF is being shutdown and that all links in the router LSA should be advertised with the value 0xFFFF and the metric value for all external and summary LSAs is set to 0xFF0000 until OSPF is restarted. This configuration will not be saved.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# max-metric router-lsa external-lsa summary-lsa
link all
```

The following command indicates that OSPF is being shutdown and that all links in the router LSA should be advertised with the value 0xFFFF and the metric value for all external and summary LSAs should be set to 0xFF0000 until OSPF is restarted. Also, if OSPF TE is enabled then all LINK TLVs advertised by the router in Opaque LSAs should be updated with the TE Metric set to 0xFFFFFFFF and the available bandwidth set to 0. This configuration will not be saved.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# max-metric router-lsa all-lsas link all
```

The following command indicates that OSPF is being shutdown and that all links in the router LSA should be advertised with the value 0xFFFF and the metric value for all summary LSAs should be set to 0xFFFFFE until OSPF is restarted. This configuration will not be saved.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# max-metric router-lsa summary-lsa 16777214 link
all
```

The following command turns off the advertisement of special metric values in all Router, Summary, and External LSAs.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# no max-metric router-lsa
```

Configuring OSPF shortest path first throttling

To set OSPF shortest path first throttling to the values in the previous example, use the following command.

```
NetIron(configure)# timer throttle spf 200 300 2000
```

Syntax: [no] timer throttle spf <*initial-delay*> <*hold-time*> <*max-hold-time*>

The <*initial-delay*> variable sets the initial value for the SPF delay in milliseconds. Possible values are between 0 and 65535 milliseconds.

The <*hold-time*> variable sets the minimum hold time between SPF calculations after the initial delay. This value will be doubled after hold-time expires until the max-hold-time is reached. Possible values are between 0 and 65535 milliseconds.

The <*max-hold-time*> variable sets the maximum hold time between SPF calculations Possible values are between 0 and 65535 milliseconds.

NOTE
The hold time values that you specify are rounded up to the next highest 100 ms value. For example, any value between 0 and 99 will be configured as 100 ms.

Command replacement

This command overlaps in functionality with the timer throttle spf command which will be phased out from the Multi-Service IronWare software. To use this command to replicate the exact functionality of the timer throttle spf command configure it as shown in the following.

```
NetIron(configure)# timer throttle spf 1000 5000 5000
```

Displaying OSPF Router Advertisement

Using the **show ip ospf** command you can display the current OSPF Router Advertisement configuration. The text show below in bold is displayed for an OSPF Router Advertisement configuration.

```
NetIron#show ip ospf
OSPF Version                 Version 2
Router Id                    10.10.10.10
ASBR Status                  No
ABR Status                   No         (0)
Redistribute Ext Routes from
External LSA Counter         5
External LSA Checksum Sum    0002460e
Originate New LSA Counter    5
Rx New LSA Counter           8
External LSA Limit           14447047
Database Overflow Interval   0
Database Overflow State :    NOT OVERFLOWED
RFC 1583 Compatibility :     Enabled
Originating router-LSAs with maximum metric
   Condition: Always Current State: Active
   Link Type: PTP STUB TRANSIT
   Additional LSAs originated with maximum metric:
     LSA Type              Metric Value
     AS-External             16711680
```

```
Type 3 Summary          16711680
Type 4 Summary          16711680
Opaque-TE               4294967295
```

Displaying OSPF information

You can display the following OSPF information:

- Trap, area, and interface information – refer to "Displaying general OSPF configuration information" on page 911.
- CPU utilization statistics – refer to "Displaying CPU utilization and other OSPF tasks" on page 912.
- Area information – refer to "Displaying OSPF area information" on page 913.
- Neighbor information – refer to "Displaying OSPF neighbor information" on page 914.
- Interface information – refer to "Displaying OSPF interface information" on page 916.
- Route information – refer to "Displaying OSPF route information" on page 919.
- External link state information – refer to "Displaying OSPF external link state information" on page 923.
- Database Information – refer to "Displaying OSPF database information" on page 921
- Link state information – refer to "Displaying OSPF database link state information" on page 924.
- Virtual Neighbor information – refer to "Displaying OSPF virtual neighbor and link information" on page 928.
- Virtual Link information – refer to "Displaying OSPF virtual link information" on page 929.
- ABR and ASBR information – refer to "Displaying OSPF ABR and ASBR information" on page 925.
- Trap state information – refer to "Displaying OSPF trap status" on page 926.
- OSPF Point-to-Point Links – refer to "Viewing Configured OSPF point-to-point links" on page 926.
- OSPF Graceful Restart information refer to "Displaying an OSPF Graceful Restart information" on page 930
- OSPF Router Advertisement information refer to "Displaying OSPF Router Advertisement information" on page 931

Displaying general OSPF configuration information

To display general OSPF configuration information, enter the following command at any CLI level.

```
NetIron# show ip ospf config
Router OSPF: Enabled
Redistribution: Disabled
Default OSPF Metric: 10
OSPF Redistribution Metric: Type2

OSPF External LSA Limit: 1447047

OSPF Database Overflow Interval: 0

RFC 1583 Compatibility: Enabled

Router id: 207.95.11.128

Interface State Change Trap:                      Enabled
Virtual Interface State Change Trap:              Enabled
Neighbor State Change Trap:                       Enabled
Virtual Neighbor State Change Trap:               Enabled
Interface Configuration Error Trap:               Enabled
Virtual Interface Configuration Error Trap:       Enabled
Interface Authentication Failure Trap:            Enabled
Virtual Interface Authentication Failure Trap:    Enabled
Interface Receive Bad Packet Trap:                Enabled
Virtual Interface Receive Bad Packet Trap:        Enabled
Interface Retransmit Packet Trap:                 Disabled
Virtual Interface Retransmit Packet Trap:         Disabled
Originate LSA Trap:                               Disabled
Originate MaxAge LSA Trap:                        Disabled
Link State Database Overflow Trap:                Disabled
Link State Database Approaching Overflow Trap:    Disabled

OSPF Area currently defined:
Area-ID          Area-Type Cost
0                normal    0

OSPF Interfaces currently defined:
Ethernet Interface: 3/1-3/2
ip ospf md5-authentication-key-activation-wait-time 300
ip ospf cost 0
ip ospf area 0

Ethernet Interface: v1
ip ospf md5-authentication-key-activation-wait-time 300
ip ospf cost 0
ip ospf area 0
```

Syntax: show ip ospf config

Displaying CPU utilization and other OSPF tasks

You can display CPU utilization statistics for OSPF and other tasks.

To display CPU utilization statistics, enter the following command.

```
NetIron#show tasks
Task Name   Pri   State     PC     Stack  Size  CPU Usage(%) task id  task vid
----------  ---  -----  ---------  --------  -----  ---------  ------ -------
       idle 0    ready  00001904  04058fa0  4096    99          0        0
    monitor 20    wait  0000d89c  0404bd80  8192    0           0        0
        int 16    wait  0000d89c  04053f90  16384   0           0        0
      timer 15    wait  0000d89c  04057f90  16384   0           0        0
        dbg 30    wait  0000d89c  0404ff08  8192    0           0        0
      flash 17    wait  0000d89c  0409ff90  8192    0           0        0
         wd 31    wait  0000d89c  0409df80  8192    0           0        0
       boot 17    wait  0000d89c  04203e28  65536   0           0        0
       main 3     wait  0000d89c  2060cf38  65536   0           0        1
        itc 6     wait  0000d89c  20612ae8  16384   0           0        1
        tmr 5     wait  0000d89c  20627628  16384   0           0        1
      ip_rx 5     wait  0000d89c  2062ff48  16384   0           0        1
        scp 5     wait  0000d89c  20635628  16384   0           0        1
    console 5     wait  0000d89c  2063e618  32768   0           0        1
       vlan 5     wait  0000d89c  20648618  16384   0           0        1
    mac_mgr 5     wait  0000d89c  20657628  16384   0           0        1
    mrp_mgr 5     wait  0000d89c  2065c628  16384   0           0        1
       vsrp 5     wait  0000d89c  20663620  16384   0           0        1
       snms 5     wait  0000d89c  20667628  16384   0           0        1
        rtm 5     wait  0000d89c  20674628  16384   0           0        1
       rtm6 5     wait  0000d89c  2068a628  16384   0           0        1
      ip_tx 5    ready  0000d89c  206a9628  16384   0           0        1
        rip 5     wait  0000d89c  20762628  16384   0           0        1
        bgp 5     wait  0000d89c  207e6628  16384   0           0        1
     bgp_io 5     wait  0000d89c  2082ef00  16384   0           0        1
       ospf 5     wait  0000d89c  20832628  16384   1           0        1
ospf_r_calc 5     wait  0000d89c  2089ff10  16384   0           0        1
  isis_task 5     wait  0000d89c  208a3628  16384   0           0        1
   isis_spf 5     wait  0000d89c  208a8f10  16384   0           0        1
       mcast 5     wait  0000d89c  208ac628  16384   0           0        1
        vrrp 5     wait  0000d89c  208b4628  16384   0           0        1
      ripng 5     wait  0000d89c  208b9628  16384   0           0        1
      ospf6 5     wait  0000d89c  208c3628  16384   0           0        1
   ospf6_rt 5     wait  0000d89c  208c7f08  16384   0           0        1
     mcast6 5     wait  0000d89c  208cb628  16384   0           0        1
         l4 5     wait  0000d89c  208cf620  16384   0           0        1
        stp 5     wait  0000d89c  209a7620  16384   0           0        1
       snmp 5     wait  0000d89c  209c3628  32768   0           0        1
       rmon 5     wait  0000d89c  209cc628  32768   0           0        1
        web 5     wait  0000d89c  209d6628  32768   0           0        1
       lacp 5     wait  0000d89c  209da628  16384   0           0        1
      dot1x 5     wait  0000d89c  209e0620  16384   0           0        1
  hw_access 5     wait  0000d89c  209e6628  16384   0           0        1
```

Syntax: show tasks

The displayed information shows the following:

TABLE 140 CLI display of show tasks

This field...	Displays...
Task Name	Name of task running on the device.
Pri	Priority of the task in comparison to other tasks
State	Current state of the task
PC	current instruction for the task
Stack	Stack location for the task
Size	Stack size of the task
CPU Usage(%)	Percentage of the CPU being used by the task
task id	Task's ID number assigned by the operating system.
task vid	A memory domain ID.

Displaying OSPF area information

To display OSPF area information, enter the following command at any CLI level.

```
NetIron# show ip ospf area
Indx  Area       Type   Cost  SPFR  ABR  ASBR  LSA  Chksum(Hex)
1   0.0.0.0      normal  0    1      0     0    1    0000781f
2  192.147.60.0  normal  0    1      0     0    1    0000fee6
3  192.147.80.0  stub    1    1      0     0    2    000181cd
```

Syntax: show ip ospf area [<*area-id*>] | [<*num*>]

The <*area-id*> parameter shows information for the specified area.

The <*num*> parameter displays the entry that corresponds to the entry number you enter. The entry number identifies the entry's position in the area table.

This display shows the following information.

TABLE 141 CLI display of OSPF area information

This field...	Displays...
Indx	The row number of the entry in the router's OSPF area table.
Area	The area number.
Type	The area type, which can be one of the following: • nssa • normal • stub
Cost	The area's cost.
SPFR	The SPFR value.
ABR	The ABR number.
ASBR	The ABSR number.

TABLE 141 CLI display of OSPF area information (Continued)

This field...	Displays...
LSA	The LSA number.
Chksum(Hex)	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.

Displaying OSPF neighbor information

To display OSPF neighbor information, enter the following command at any CLI level.

```
NetIron# show ip ospf neighbor

Port  Address         Pri  State      Neigh Address   Neigh ID     Ev Op Cnt
v10   10.1.10.1       1    FULL/DR    10.1.10.2       10.65.12.1    5  2   0
v11   10.1.11.1       1    FULL/DR    10.1.11.2       10.65.12.1    5  2   0
v12   10.1.12.1       1    FULL/DR    10.1.12.2       10.65.12.1    5  2   0
v13   10.1.13.1       1    FULL/DR    10.1.13.2       10.65.12.1    5  2   0
v14   10.1.14.1       1    FULL/DR    10.1.14.2       10.65.12.1    5  2   0
```

Syntax: show ip ospf neighbor [router-id <*ip-addr*> **|** <*num*> **| extensive]**

The **router-id** <*ip-addr*> parameter displays only the neighbor entries for the specified router.

The <*num*> parameter displays only the entry in the specified index position in the neighbor table. For example, if you enter "1", only the first entry in the table is displayed.

The **extensive** option displays detailed information about the neighbor.

These displays show the following information.

TABLE 142 CLI display of OSPF neighbor information

Field	Description
Port	The port through which the device is connected to the neighbor.
Address	The IP address of the port on which this device is connected to the neighbor.
Pri	The OSPF priority of the neighbor. • For multi-access networks, the priority is used during election of the Designated Router (DR) and Backup designated Router (BDR). • For point-to-point links, this field shows one of the following values: • 1 = point-to-point link • 3 = point-to-point link with assigned subnet

TABLE 142 CLI display of OSPF neighbor information (Continued)

Field	Description
State	The state of the conversation between the device and the neighbor. This field can have one of the following values: • Down – The initial state of a neighbor conversation. This value indicates that there has been no recent information received from the neighbor. • Attempt – This state is only valid for neighbors attached to non-broadcast networks. It indicates that no recent information has been received from the neighbor. • Init – A Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor. (The router itself did not appear in the neighbor's Hello packet.) All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface. • 2-Way – Communication between the two routers is bidirectional. This is the most advanced state before beginning adjacency establishment. The Designated Router and Backup Designated Router are selected from the set of neighbors in the 2-Way state or greater. • ExStart – The first step in creating an adjacency between the two neighboring routers. The goal of this step is to decide which router is the master, and to decide upon the initial Database Description (DD) sequence number. Neighbor conversations in this state or greater are called adjacencies. • Exchange – The router is describing its entire link state database by sending Database Description packets to the neighbor. Each Database Description packet has a DD sequence number, and is explicitly acknowledged. Only one Database Description packet can be outstanding at any time. In this state, Link State Request packets can also be sent asking for the neighbor's more recent advertisements. All adjacencies in Exchange state or greater are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • Loading – Link State Request packets are sent to the neighbor asking for the more recent advertisements that have been discovered (but not yet received) in the Exchange state. • Full – The neighboring routers are fully adjacent. These adjacencies will now appear in router links and network link advertisements.
Neigh Address	The IP address of the neighbor. For point-to-point links, the value is as follows: • If the **Pri** field is "1", this value is the IP address of the neighbor router's interface. • If the **Pri** field is "3", this is the subnet IP address of the neighbor router's interface.
Neigh ID	The neighbor router's ID.
Ev	The number of times the neighbor's state changed.
Opt	The sum of the option bits in the Options field of the Hello packet. This information is used by Dell technical support. Refer to Section A.2 in RFC 2178 for information about the Options field in Hello packets.
Cnt	The number of LSAs that were retransmitted.

Displaying OSPF interface information

To display OSPF interface information, enter the following command at any CLI level.

```
NetIron# show ip ospf interface ethernet 1/11

Ethernet 1/11 admin up, oper up
     IP Address 15.1.1.15, Area 0
     Database Filter: Not Configured
     State BDR, Pri 1, Cost 1, Options 2, Type broadcast Events 2
    Timers(sec): Transmit 1, Retrans 5, Hello 10, Dead 40
     DR:  Router ID 192.168.254.1     Interface Address 15.1.1.1
     BDR: Router ID 10.0.0.15         Interface Address 15.1.1.15
     Neighbor Count = 1, Adjacent Neighbor Count= 1
     Neighbor:         15.1.1.1 (DR)
     Authentication-Key: None
     MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

If you specify an interface that is not configured within a specified VRF, then the following error message will display as shown in the example below:

```
NetIron# show ip ospf vrf one interface ethernet 1/1
Error: Interface(eth 1/1) not part of VRF(one)
```

NOTE
You cannot display multiple ports for any interfaces. For example, when displaying OSPF interface information on ethernet 1/1 only one port can displayed at a given time.

Syntax: **show ip ospf [vrf** <*vrf-name*>**] interface [**<*ip-addr*>**] [brief] [ethernet** <*port*> **| pos** <*port*> **| loopback** <*number*> **| tunnel** <*number*> **| ve** <*number*> **]**

The [**vrf** <*vrf-name*>] parameter displays information for VRF, or a specific vrf-name.

The [<*ip-addr*>] parameter displays the OSPF interface information for the specified IP address.

The [**brief**] parmeter displays interface information in the brief mode. Refer to "Displaying OSPF interface brief information" on page 918.

The **ethernet | pos | loopback | tunnel | ve** parameter specifies the interface for which to display information. If you specify an Ethernet interface or a POS interface, you can also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, you can also specify the number associated with the interface.

The following table defines the highlighted fields shown in the above example output of the **show ip ospf interface ethernet** command.

TABLE 143 Output of the **show ip ospf interface** command

This field	Displays
Interface	The type of interface type and the port number or number of the interface.
IP Address	The IP address of the interface.
Area	The OSPF area configured on the interface
Database Filter	The router's configuration for blocking outbound LSAs on an OSPF interface as described in "Block flooding of outbound LSAs on specific OSPF interfaces" on page 878. If Not Configured is displayed, there is no outbound LSA filter configured. This is the default condition.

TABLE 143 Output of the **show ip ospf interface** command (Continued)

This field	Displays
State	The state of the interface. Possible states include the following: • DR – The interface is functioning as the Designated Router for OSPFv2. • BDR – The interface is functioning as the Backup Designated Router for OSPFv2. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.
Pri	The link ID as defined in the router-LSA. This value can be one of the following: 1 = point-to-point link 3 = point-to-point link with an assigned subnet
Cost	The configured output cost for the interface.
Options	OSPF Options (Bit7 - Bit0): • unused:1 • opaque:1 • summary:1 • dont_propagate:1 • nssa:1 • multicast:1 • externals:1 • tos:1
Type	The area type, which can be one of the following: • Broadcast • Point to Point • non-broadcast • Virtual Link
Events	OSPF Interface Event: • Interface_Up = 0x00 • Wait_Timer = 0x01 • Backup_Seen = 0x02 • Neighbor_Change = 0x03 • Loop_Indication = 0x04 • Unloop_Indication = 0x05 • Interface_Down = 0x06 • Interface_Passive = 0x07
Timer intervals	The interval, in seconds, of the transmit-interval, retransmit-interval, hello-interval, and dead-interval timers.
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.
Neighbor Count	The number of neighbors to which the interface is connected.
Adjacent Neighbor Count	The number of adjacent neighbor routers.
Neighbor:	The IP address of the neighbor.

Displaying OSPF interface brief information

The following command is used to display the OSPF database brief information.

```
NetIron# show ip ospf interface brief
Number of Interfaces is 1
Interface Area IP Addr/Mask Cost State Nbrs(F/C)
eth 1/2    0    16.1.1.2/24   1     down  0/0
```

Table 144 defines the fields shown in the above example output of the **show ip ospf interface brief** command.

TABLE 144 Output of the **show ip ospf interface brief** command

This field	Displays
Interface	The interface through which the router is connected to the neighbor.
Area	The OSPF Area that the interface is configured in.
IP Addr/Mask	The IP address and mask of the interface.
Cost	The configured output cost for the interface.
State	The state of the conversation between the router and the neighbor. This field can have one of the following values: • Down – The initial state of a neighbor conversation. This value indicates that there has been no recent information received from the neighbor. • Attempt – This state is only valid for neighbors attached to non-broadcast networks. It indicates that no recent information has been received from the neighbor. • Init – A Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor. (The router itself did not appear in the neighbor's Hello packet.) All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface. • 2-Way – Communication between the two routers is bidirectional. This is the most advanced state before beginning adjacency establishment. The Designated Router and Backup Designated Router are selected from the set of neighbors in the 2-Way state or greater. • ExStart – The first step in creating an adjacency between the two neighboring routers. The goal of this step is to decide which router is the master, and to decide upon the initial Database Description (DD) sequence number. Neighbor conversations in this state or greater are called adjacencies. • Exchange – The router is describing its entire link state database by sending Database Description packets to the neighbor. Each Database Description packet has a DD sequence number, and is explicitly acknowledged. Only one Database Description packet can be outstanding at any time. In this state, Link State Request packets can also be sent asking for the neighbor's more recent advertisements. All adjacencies in Exchange state or greater are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • Loading – Link State Request packets are sent to the neighbor asking for the more recent advertisements that have been discovered (but not yet received) in the Exchange state. • Full – The neighboring routers are fully adjacent. These adjacencies will now appear in router links and network link advertisements.
Nbrs(F/C)	The number of adjacent neighbor routers. The number to the left of the "/" are the neighbor routers that are fully adjacent and the number to the right represents all adjacent neighbor routers.

Displaying OSPF route information

To display OSPF route information, enter the following command at any CLI level.

```
NetIron#show ip ospf route
OSPF Area 0x00000000 ASBR Routes 1:

      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.65.12.1      255.255.255.255 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.65.12.1      10.65.12.1      Asbr      Valid      0          6000
      Paths Out_Port  Next_Hop        Type      State
      1     v49       10.1.49.2       OSPF      21 01
      2     v12       10.1.12.2       OSPF      21 01
      3     v11       10.1.11.2       OSPF      21 01
      4     v10       10.1.10.2       OSPF      00 00
OSPF Area 0x00000041 ASBR Routes 1:

      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.65.12.1      255.255.255.255 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.65.12.1      10.65.12.1      Asbr      Valid      0          6000
      Paths Out_Port  Next_Hop        Type      State
      1     v204      10.65.5.251     OSPF      21 01
      2     v201      10.65.2.251     OSPF      20 d1
      3     v202      10.65.3.251     OSPF      20 cd
      4     v205      10.65.6.251     OSPF      00 00
OSPF Area Summary Routes 1:

      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.65.0.0       255.255.0.0     0         0          Inter
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       0.0.0.0         Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     1/1       0.0.0.0         DIRECT    00 00
OSPF Regular Routes 208:

      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.1.10.0       255.255.255.252 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       10.1.10.2       Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     v10       0.0.0.0         OSPF      00 00

      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.1.11.0       255.255.255.252 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       10.1.11.2       Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     v11       0.0.0.0         OSPF      00 00
```

Syntax: show ip ospf routes [*<ip-addr>*]

The *<ip-addr>* parameter specifies a destination IP address. If you use this parameter, only the route entries for that destination are shown.

This display shows the following information.

TABLE 145 CLI display of OSPF route information

This field...	Displays...
Destination	The IP address of the route's destination.
Mask	The network mask for the route.
Path_Cost	The cost of this route path. (A route can have multiple paths. Each path represents a different exit port for the device.)
Type2_Cost	The type 2 cost of this path.
Path_Type	The type of path, which can be one of the following: • Inter – The path to the destination passes into another area. • Intra – The path to the destination is entirely within the local area. • External1 – The path to the destination is a type 1 external route. • External2 – The path to the destination is a type 2 external route.
Adv_Router	The OSPF router that advertised the route to this device.
Link-State	The link state from which the route was calculated.
Dest_Type	The destination type, which can be one of the following: • ABR – Area Border Router • ASBR – Autonomous System Boundary Router • Network – the network
State	The route state, which can be one of the following: • Changed • Invalid • Valid This information is used by Dell technical support.
Tag	The external route tag.
Flags	State information for the route entry. This information is used by Dell technical support.
Paths	The number of paths to the destination.
Out_Port	The router port through which the device reaches the next hop for this route path.
Next_Hop	The IP address of the next-hop router for this path.
Type	The route type, which can be one of the following: • OSPF • Static Replaced by OSPF
State	State information for the path. This information is used by Dell technical support.

Displaying the routes that have been redistributed into OSPF

You can display the routes that have been redistributed into OSPF. To display the redistributed routes, enter the following command at any level of the CLI.

```
NetIron# show ip ospf redistribute route
  4.3.0.0 255.255.0.0 static
  3.1.0.0 255.255.0.0 static
  10.11.61.0 255.255.255.0 connected
  4.1.0.0 255.255.0.0 static
```

In this example, four routes have been redistributed. Three of the routes were redistributed from static IP routes and one route was redistributed from a directly connected IP route.

Syntax: show ip ospf redistribute route [<*ip-addr*> <*ip-mask*>]

The <*ip-addr*> <*ip-mask*> parameter specifies a network prefix and network mask. Here is an example.

```
NetIron# show ip ospf redistribute route 3.1.0.0 255.255.0.0
  3.1.0.0 255.255.0.0 static
```

Displaying OSPF database information

The following command is used to display the OSPF database.

```
NetIron#show ip ospf database
Graceful Link States
Area  Interface  Adv Rtr  Age Seq(Hex) Prd Rsn  Nbr Intf IP
0     eth 1/2    2.2.2.2  7   80000001 60  SW   6.1.1.2

Router Link States
Index AreaID            Type LS ID      Adv Rtr     Seq(Hex)   Age  Cksum
1     0                 Rtr  2.2.2.2    2.2.2.2     80000003   93   0xac6c
2     0                 Rtr  1.1.1.1    1.1.1.1     80000005   92   0x699e
3     0                 Net  16.1.1.2   2.2.2.2     80000002   93   0xbd73
4     0                 OpAr 1.0.0.3    1.1.1.1     80000005   83   0x48e7
5     0                 OpAr 1.0.0.2    2.2.2.2     80000006   80   0x50da
6     111.111.111.111   Rtr  1.1.1.1    1.1.1.1     80000004   142  0x0a38
7     111.111.111.111   Summ 1.1.1.1    1.1.1.1     80000001   147  0x292b
8     111.111.111.111   OpAr 1.0.0.2    1.1.1.1     80000002   179  0x063f

Type-5 AS External Link States
Index Age  LS ID     Router   Netmask   Metric    Flag  Fwd Address
1     147  9.9.1.13  1.1.1.1  ffffffff  0000000a  0000  0.0.0.0
2     147  9.9.1.26  1.1.1.1  ffffffff  0000000a  0000  0.0.0.0
```

Syntax: show ip ospf database

This display shows the information described in Table 146.

TABLE 146 CLI display of OSPF database information

This field...	Displays...
Area	The OSPF area that the interface configured for OSPF graceful restart is in.
Interface	The interface that is configured for OSPF graceful restart.

TABLE 146 CLI display of OSPF database information (Continued)

This field...	Displays...
Prd	Grace Period: The number of seconds that the router's neighbors should continue to advertise the router as fully adjacent, regardless of the state of database synchronization between the router and its neighbors. Since this time period began when grace-LSA's LS age was equal to 0, the grace period terminates when either: • the LS age of the grace-LSA exceeds the value of a Grace Period • the grace-LSA is flushed.
Rsn	Graceful restart reason: The reason for the router restart defined as one of the following: • UK – unknown • RS – software restart • UP – software upgrade or reload • SW – switch to redundant control processor
Nbr Intf IP	The IP address of the OSPF graceful restart neighbor .
Index	ID of the entry
Aging	The age of the LSA, in seconds.
Area ID	ID of the OSPF area
Type	Link state type of the route.
LS ID	The ID of the link-state advertisement from which the router learned this route.
Adv Rtr	ID of the advertised route.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps the LSA with a sequence number. This number enables the device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA in seconds.
Chksum	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.
Router	The router IP address.
Netmask	The subnet mask of the network.
Metric	The cost (value) of the route
Flag	State information for the route entry. This information is used by Dell technical support.

Displaying OSPF external link state information

To display external link state information, enter the following command at any CLI level.

```
NetIron#show ip ospf database external-link-state
Index Aging  LS ID           Router          Netmask  Metric    Flag
1     591    10.65.13.0      10.65.12.1      ffffff00 8000000a  0000
2     591    10.65.16.0      10.65.12.1      ffffff00 8000000a  0000
3     591    10.65.14.0      10.65.12.1      ffffff00 8000000a  0000
4     591    10.65.17.0      10.65.12.1      ffffff00 8000000a  0000
5     592    10.65.12.0      10.65.12.1      ffffff00 8000000a  0000
6     592    10.65.15.0      10.65.12.1      ffffff00 8000000a  0000
7     592    10.65.18.0      10.65.12.1      ffffff00 8000000a  0000
```

Syntax: **show ip ospf database external-link-state [advertise** <*num*> **| extensive | link-state-id** <*ip-addr*> **| router-id** <*ip-addr*> **| sequence-number** <*num(Hex)*> **]**

The **advertise** <*num*> parameter displays the hexadecimal data in the specified LSA packet. The <*num*> parameter identifies the LSA packet by its position in the router's External LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf external-link-state** command to display the table.

The **extensive** option displays the LSAs in decrypted format.

NOTE
You cannot use the **extensive** option in combination with other display options. The entire database is displayed.

The **link-state-id** <*ip-addr*> parameter displays the External LSAs for the LSA source specified by <*IP-addr*>.

The **router-id** <*ip-addr*> parameter shows the External LSAs for the specified OSPF router.

The **sequence-number** <*num(Hex)*> parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

This display shows the following information.

TABLE 147 CLI display of OSPF external link state information

This field...	Displays...
Index	ID of the entry
Aging	The age of the LSA, in seconds.
LS ID	The ID of the link-state advertisement from which the device learned this route.
Router	The router IP address.
Netmask	The subnet mask of the network.
Metric	The cost (value) of the route
Flag	State information for the route entry. This information is used by Dell technical support.

Displaying OSPF database-summary information

To display database-summary information, enter the following command at any CLI level.

```
MLXe2#show ip ospf database database-summary
Area ID      Router  Network Sum-Net  Sum-ASBR  NSSA-Ext  Opq-Area  Subtotal
0.0.0.0      104      184     19       42        0         0         349
AS External                                                         308
Total       104     184     19      42       0         0        657
```

Syntax: show ip ospf database database-summary

TABLE 148 CLI display of OSPF database summary information

This field...	Displays...
Area ID	The area number.
Router	The number of router link state advertisements in that area.
Network	The number of network link state advertisements in that area.
Sum-Net	The number of summary link state advertisements in that area.
Sum-ASBR	The number of summary autonomous system boundary router (ASBR) link state advertisements in that area
NSSA-Ext	The number of not-so-stubby
Opq-area	the number of Type-10 (area-scope) Opaque LSA.s

Displaying OSPF database link state information

To display database link state information, enter the following command.

```
NetIron# show ip ospf database link-state
Index Area ID        Type   LS ID           Adv Rtr          Seq(Hex) Age  Cksum
1     0              Rtr  10.1.10.1       10.1.10.1       800060ef 3    0x4be2
2     0              Rtr  10.65.12.1      10.65.12.1      80005264 6    0xc870
3     0              Net  10.1.64.2       10.65.12.1      8000008c 1088 0x06b7
4     0              Net  10.1.167.2      10.65.12.1      80000093 1809 0x86c8
5     0              Net  10.1.14.2       10.65.12.1      8000008c 1088 0x2ec1
6     0              Net  10.1.117.2      10.65.12.1      8000008c 1087 0xbccb
7     0              Net  10.1.67.2       10.65.12.1      8000008c 1088 0xe4d5
8     0              Net  10.1.170.2      10.65.12.1      80000073 604  0xa5c6
9     0              Net  10.1.17.2       10.65.12.1      8000008c 1088 0x0ddf
10    0              Net  10.1.120.2      10.65.12.1      8000008c 1087 0x9be9
11    0              Net  10.1.70.2       10.65.12.1      8000008c 1088 0xc3f3
12    0              Net  10.1.173.2      10.65.12.1      80000017 1087 0x3d88
13    0              Net  10.1.20.2       10.65.12.1      8000008c 1088 0xebfd
14    0              Net  10.1.123.2      10.65.12.1      8000008c 1087 0x7a08
15    0              Net  10.1.73.2       10.65.12.1      8000008c 1088 0xa212
16    0              Net  10.1.176.2      10.65.12.1      80000025 1087 0xffb4
17    0              Net  10.1.23.2       10.65.12.1      8000008c 1088 0xca1c
18    0              Net  10.1.126.2      10.65.12.1      8000008c 1087 0x5926
```

Syntax: **show ip ospf database link-state** [**advertise** <*num*> | **asbr** [<*ip-addr*>] [**adv-router** <*ip-addr*>] | **extensive** | **link-state-id** <*ip-addr*> | **network** [<*ip-addr*>] [**adv-router** <*ip-addr*>] | **nssa** [<*ip-addr*>] [**adv-router** <*ip-addr*>] | **router** [<*ip-addr*>] [**adv-router** <*ip-addr*>] | **router-id** <*ip-addr*> | **self-originate l sequence-number** <*num(Hex)*> | **summary** [<*ip-addr*>] [**adv-router** <*ip-addr*>]

The **advertise** <*num*> parameter displays the hexadecimal data in the specified LSA packet. The <*num*> parameter identifies the LSA packet by its position in the router's LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf link-state** command to display the table.

The **asbr** option shows ASBR LSAs.

The **extensive** option displays the LSAs in decrypted format.

NOTE
You cannot use the **extensive** option in combination with other display options. The entire database is displayed.

The **link-state-id** <*ip-addr*> parameter displays the LSAs for the LSA source specified by <*IP-addr*>.

The **network** option shows network LSAs.

The **nssa** option shows NSSA LSAs.

The **router-id** <*ip-addr*> parameter shows the LSAs for the specified OSPF router.

The **sequence-number** <*num(Hex)*> parameter displays the LSA entries for the specified hexadecimal LSA sequence number.

The **self-originate** option shows self-originated LSAs.

The **summary** option shows summary information.

TABLE 149 CLI display of OSPF database link state information

This field...	Displays...
Index	ID of the entry
Area ID	ID of the OSPF area
Type LS ID	Link state type of the route
Adv Rtr	ID of the advertised route
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps the LSA with a sequence number. This number enables the device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA in seconds.
Cksum	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.

Displaying OSPF ABR and ASBR information

To display OSPF ABR and ASBR information, enter the following command at any CLI level.

```
NetIron# show ip ospf border-routers
```

Syntax: show ip ospf border-routers [<*ip-addr*>]

The <*ip-addr*> parameter displays the ABR and ASBR entries for the specified IP address.

```
NetIron# show ip ospf border-routers

      router ID       router type next hop router  outgoing interface  Area
1     10.65.12.1      ABR         10.1.49.2        v49                 0
1     10.65.12.1      ASBR        10.1.49.2        v49                 0
1     10.65.12.1      ABR         10.65.2.251      v201                65
1     10.65.12.1      ASBR        10.65.2.251      v201                65
```

Syntax: show ip ospf border-routers

TABLE 150 CLI display of OSPF border routers

This field...	Displays...
(Index)	Displayed index number of the border router.
Router ID	ID of the OSPF router
Router type	Type of OSPF router: ABR or ASBR
Next hop router	ID of the next hop router
Outgoing interface	ID of the interface on the router for the outgoing route.
Area	ID of the OSPF area to which the OSPF router belongs

Displaying OSPF trap status

All traps are enabled by default when you enable OSPF. To disable or re-enable an OSPF trap, refer to "Modify OSPF traps generated" on page 902.

To display the state of each OSPF trap, enter the following command at any CLI level.

```
NetIron# show ip ospf trap
Interface State Change Trap:                      Enabled
Virtual Interface State Change Trap:              Enabled
Neighbor State Change Trap:                       Enabled
Virtual Neighbor State Change Trap:               Enabled
Interface Configuration Error Trap:               Enabled
Virtual Interface Configuration Error Trap:       Enabled
Interface Authentication Failure Trap:            Enabled
Virtual Interface Authentication Failure Trap:    Enabled
Interface Receive Bad Packet Trap:                Enabled
Virtual Interface Receive Bad Packet Trap:        Enabled
Interface Retransmit Packet Trap:                 Disabled
Virtual Interface Retransmit Packet Trap:         Disabled
Originate LSA Trap:                               Disabled
Originate MaxAge LSA Trap:                        Disabled
Link State Database Overflow Trap:                Disabled
Link State Database Approaching Overflow Trap:    Disabled
```

Syntax: show ip ospf trap

Viewing Configured OSPF point-to-point links

You can use the show ip ospf interface command to display OSPF point-to-point information. Enter the following command at any CLI level.

```
NetIron# show ip ospf interface 192.168.1.1
```

```
Ethernet 2/1,OSPF enabled
IP Address 192.168.1.1, Area 0
OSPF state ptr2ptr, Pri 1, Cost 1, Options 2, Type pt-2-pt Events 1
Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
DR: Router ID 0.0.0.0 Interface Address 0.0.0.0
BDR: Router ID 0.0.0.0 Interface Address 0.0.0.0
Neighbor Count = 0, Adjacent Neighbor Count= 1
Neighbor: 2.2.2.2
Authentication-Key:None
MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

Syntax: show ip ospf interface [<*ip-addr*>]

The <*ip-addr*> parameter displays the OSPF interface information for the specified IP address.

The following table defines the highlighted fields shown in the above example output of the show ip ospf interface command

TABLE 151 Output of the show ip ospf interface command

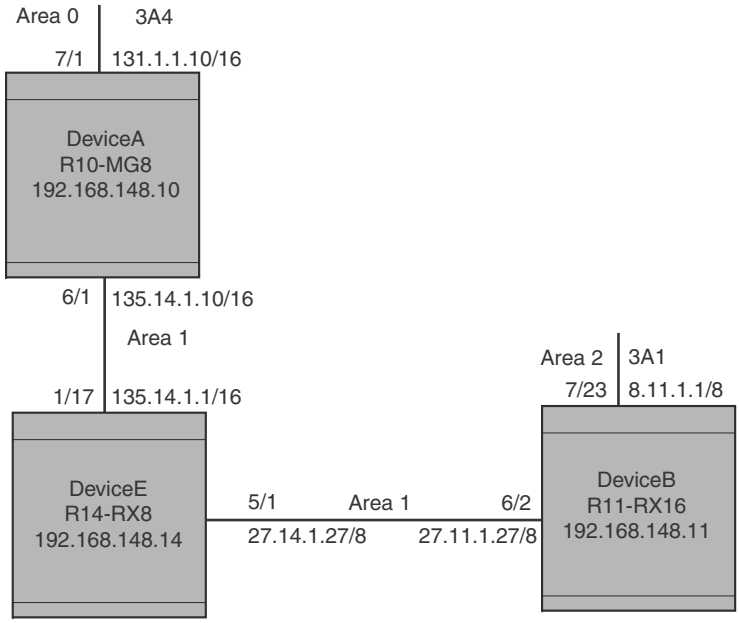
This field	Displays
IP Address	The IP address of the interface.
OSPF state	The OSPF state of the interface.
Pri	The router priority.
Cost	The configured output cost for the interface.
Options	OSPF Options (Bit7 - Bit0): • unused:1 • opaque:1 • summary:1 • dont_propagate:1 • nssa:1 • multicast:1 • externals:1 • tos:1
Type	The area type, which can be one of the following: • Broadcast = 0x01 • NBMA = 0x02 • Point to Point = 0x03 • Virtual Link = 0x04 • Point to Multipoint = 0x05
Events	OSPF Interface Event: • Interface_Up = 0x00 • Wait_Timer = 0x01 • Backup_Seen = 0x02 • Neighbor_Change = 0x03 • Loop_Indication = 0x04 • Unloop_Indication = 0x05 • Interface_Down = 0x06 • Interface_Passive = 0x07
Adjacent Neighbor Count	The number of adjacent neighbor routers.
Neighbor:	The IP address of the neighbor.

Displaying OSPF virtual neighbor and link information

You can display OSPF virtual neighbor and virtual link information. For example, the following show run display shows the configuration in Figure 140.

```
NetIron#show run
Current configuration:
!
ver V2.2.1T143
module 1 rx-bi-1g-24-port-fiber
module 2 rx-bi-10g-4-port
module 6 rx-bi-10g-4-port
module 7 rx-bi-1g-24-port-copper
!
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
!
clock summer-time
clock timezone us Pacific
hostname R11-RX8
router ospf
 area 2
 area 1
 area 1 virtual-link 131.1.1.10
```

FIGURE 140 OSPF virtual neighor and virtual link example



Displaying OSPF virtual neighbor

Use the **show ip ospf virtual neighbor** command to display OSPF virtual neighbor information. The following example relates to the configuration in Figure 140.

```
NetIron# show ip ospf virtual neighbor
Indx Transit Area    Router ID       Neighbor address options
1    1               131.1.1.10      135.14.1.10      2
     Port   Address         state           events          count
     6/2    27.11.1.27      FULL            5               0
```

Syntax: show ip ospf virtual neighbor [<*num*>]

The <*num*> parameter displays the table beginning at the specified entry number.

Displaying OSPF virtual link information

Use the **show ip ospf virtual link** command to display OSPF virtual link information. The output below represents the virtual links configured in Figure 140.

```
NetIron# show ip ospf virtual link
Indx Transit Area     Router ID        Transit(sec) Retrans(sec) Hello(sec)
1    1                131.1.1.10       1            5            10
     Dead(sec)        events           state        Authentication-Key
     40               1                ptr2ptr      None
     MD5 Authentication-Key:       None
     MD5 Authentication-Key-Id:    None
     MD5 Authentication-Key-Activation-Wait-Time:      300
```

Syntax: show ip ospf virtual link [<*num*>]

The <*num*> parameter displays the table beginning at the specified entry number.

Clearing OSPF neighbors

You can clear all OSPF neighbors or a specified OSPF neighbor using the following command.

```
NetIron# clear ip ospf neighbor all
```

Syntax: clear ip ospf neighbor all | <*ip-address*>

Selecting the **all** option clears all of the OSPF neighbors on the router.

The <*ip-address*> variable allows you to clear a specific OSPF neighbor.

Displaying an OSPF Graceful Restart information

To display OSPF Graceful Restart information for OSPF neighbors use the **show ip ospf neighbors** command as shown in the following.

```
NetIron#show ip ospf neighbors
Port  Address        Pri State       Neigh Address   Neigh ID        Ev Opt Cnt
2/7   50.50.50.10    0   FULL/OTHER  50.50.50.1      10.10.10.30     21 66  0
  < in graceful restart state,  helping 1, timer 60 sec >
```

Use the following command to display Type 9 Graceful LSAs on a router.

```
NetIron#show ip ospf database grace-link-state
Graceful Link States
Area  Interface  Adv Rtr  Age Seq(Hex) Prd Rsn  Nbr Intf IP
0     eth 1/2    2.2.2.2  7   80000001 60  SW   6.1.1.2
```

This display shows the following information.

TABLE 152 CLI display of OSPF database grace link state information

This field...	Displays...
Area	The OSPF area that the interface configured for OSPF graceful restart is in.
Interface	The interface that is configured for OSPF graceful restart.
Adv Rtr	ID of the advertised route.
Age	The age of the LSA in seconds.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps the LSA with a sequence number. This number enables the device and other OSPF routers to determine which LSA for a given route is the most recent.
Prd	Grace Period: The number of seconds that the router's neighbors should continue to advertise the router as fully adjacent, regardless of the state of database synchronization between the router and its neighbors. Since this time period began when grace-LSA's LS age was equal to 0, the grace period terminates when either: • the LS age of the grace-LSA exceeds the value of a Grace Period • the grace-LSA is flushed.

TABLE 152 CLI display of OSPF database grace link state information (Continued)

This field...	Displays...
Rsn	Graceful restart reason: The reason for the router restart defined as one of the following: • UK – unknown • RS – software restart • UP – software upgrade or reload • SW – switch to redundant control processor
Nbr Intf IP	The IP address of the OSPF graceful restart neighbor .

Displaying OSPF Router Advertisement information

Using the **show ip ospf** command you can display the current OSPF Router Advertisement configuration. The text show below in bold is displayed for an OSPF Router Advertisement configuration.

```
NetIron# show ip ospf
OSPF Version                    Version 2
Router Id                       10.10.10.10
ASBR Status                     No
ABR Status                      No         (0)
Redistribute Ext Routes from
External LSA Counter            5
External LSA Checksum Sum       0002460e
Originate New LSA Counter       5
Rx New LSA Counter              8
External LSA Limit              14447047
Database Overflow Interval      0
Database Overflow State :       NOT OVERFLOWED
RFC 1583 Compatibility :        Enabled
Originating router-LSAs with maximum metric
   Condition: Always Current State: Active
   Link Type: PTP STUB TRANSIT
   Additional LSAs originated with maximum metric:
     LSA Type              Metric Value
     AS-External             16711680
     Type 3 Summary          16711680
     Type 4 Summary          16711680
     Opaque-TE               4294967295
```

The 03.5.00 release of the Multi-Service IronWare enhances the **show ip ospf** command to display LSAs that have been configured with a maximum metric as described in "Configuring OSPF router advertisement" on page 907 as shown above in bold.

Clearing OSPF information

You can use the **clear ip ospf** commands to clear OSPF data on an router as described in the following:

- neighbor information – refer to "Clearing OSPF neighbors" on page 932
- reset the OSPF process – "Disabling and re-enabling the OSPF process" on page 932
- clear and re-add OSPF routes – "Clearing OSPF routes" on page 932

Clearing OSPF neighbors

You can use the following command to delete and relearn all OSPF neighbors, all OSPF neighbors for a specified interface or a specified OSPF neighbor.

```
NetIron# clear ip ospf neighbor all
```

Syntax: clear ip ospf [vrf <*vrf-name*> **] neighbor all [** <*interface*> *]* | <*interface*> **|** <*ip-address*> **[** <*interface*> **]**

Selecting the **all** option without specifying an interface clears all of the OSPF neighbors on the router.

The <*interface*> variable specifies the interface that you want to clear all of the OSPF neighbors on. The following types of interfaces can be specified:

- ethernet <*slot/port*>
- tunnel <*tunnel-ID*>
- pos <*slot/port*>
- ve <*ve-ID*>

The <*ip-address*> variable allows you to clear a specific OSPF neighbor.

Disabling and re-enabling the OSPF process

You can use the following command to disable and re-enable the OSPF process on a router.

```
NetIron# clear ip ospf all
```

Syntax: clear ip ospf [vrf <*vrf-name*> **] all**

This command resets the OSPF process and brings it back up after releasing all memory used while retaining all configurations.

Clearing OSPF routes

You can use the following command to clear all OSPF routes or to clear a specific OSPF route.

```
NetIron# clear ip ospf routes all
```

Syntax: clear ip ospf [vrf <*vrf-name*> **] routes all |** <*ip-address/prefix-length*>

Selecting the **all** option resets the OSPF routes including external routes, and OSPF internal routes.

The <*ip-address/prefix-length*> variable specifies a particular route to delete and then reschedules the SPF calculation.

Configuring IS-IS (IPv4) 25

Overview

The following list displays the IPv4 IS-IS features supported by PowerConnect B-MLXe.

- IS-IS
- Level-1 Routing
- Level-2 Routing
- Restart helper-mode
- Broadcast Pseudonode
- Three-Way Handshake for Point-to-Point Adjacencies
- IS-IS PSPF Exponential back-off
- New encryption code for passwords, authentication keys, and community strings
- IS-IS Flooding
- IS-IS Point-to-Point over Ethernet
- IS-IS over a GRE IP Tunnel
- Formation of Adjacencies
- IS-IS Blackhole Avoidance (Setting the Overload Bit)
- Priority for Designated IS Election
- Limiting Access to Adjacencies With a Neighbor
- Changing the IS-IS Level on an Interface
- Disabling and Enabling Hello Padding on an Interface
- Displaying IPv4 IS-IS Information
- IS-IS SPF Scaling

The Intermediate System to Intermediate System (IS-IS) protocol is a link-state Interior Gateway Protocol (IGP) that is based on the International Standard for Organization/International Electrotechnical Commission (ISO/IEC) Open Systems Internet Networking model (OSI). In IS-IS, an intermediate system (router) is designated as either a Level 1 or Level 2 router. A Level 1 router routes traffic only within the area in which the router resides. A Level 2 router routes traffic between areas within a routing domain.

The implementation of IS-IS is based on the following specifications and draft specifications:

- ISO/IEC 10589 – “Information Technology – Telecommunication and information exchange between systems – Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connection less-mode Network Service (ISO 8473)”, 1992
- ISO/IEC 8473 – “Information processing systems – Data Communications – Protocols for providing the connectionless-mode network service”, 1988

- ISO/IEC 9542 – "Information Technology – Telecommunication and information exchange between systems – End system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connection less-mode Network Service (ISO 8473)", 1988
- RFC 1195 – "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments", 1990.
- RFC 2763 – "Dynamic Host Name Exchange Mechanism for IS-IS", 2000.
- RFC 2966 – "Domain-wide Prefix Distribution with Two-Level IS-IS", 2000
- RFC 3373 – "Three-Way Handshake for Intermediate System to Intermediate System (IS-IS) Point-to-Point Adjacencies", 2002
- Portions of the Internet Draft "IS-IS extensions for Traffic Engineering" draft-ieff-isis-traffic-02.txt (dated 2000). that describe the Extended IP reachability type-length-value (TLV type 135) and the extended Intermediate System (IS) reachability TLV (TLV type 22). These portions provide support for the wide metric version of IS-IS. No other portion is supported on Dell's implementation of IS-IS.

NOTE
The PowerConnect does not support routing of Connectionless-Mode Network Protocol (CLNP) packets. The PowerConnect uses IS-IS for TCP/IP only.

Relationship to IP route table

The IS-IS protocol has the same relationship to the PowerConnect's IP route table that OSPF has to the IP route table. The IS-IS routes are calculated and first placed in the IS-IS route table. The routes are then transferred to the IP route table.

The protocol sends the best IS-IS path for a given destination to the IP route table for comparison to the best paths from other protocols to the same destination. The CPU selects the path with the lowest administrative distance and places that path in the IP route table:

- If the path provided by IS-IS has the lowest administrative distance, then the CPU places that IS-IS path in the IP route table.
- If a path to the same destination supplied by another protocol has a lower administrative distance, the CPU installs the other protocol's path in the IP route table instead.

The ***administrative distance*** is a protocol-independent value from 1 – 255. Each path sent to the CPU, regardless of the source of the path (IS-IS, OSPF, static IP route, and so on) has an administrative distance.

Each route source has a default administrative distance. The default administrative distance for IS-IS is 115.

You can change the administrative distance for IS-IS and other routes sources.

Intermediate systems and end systems

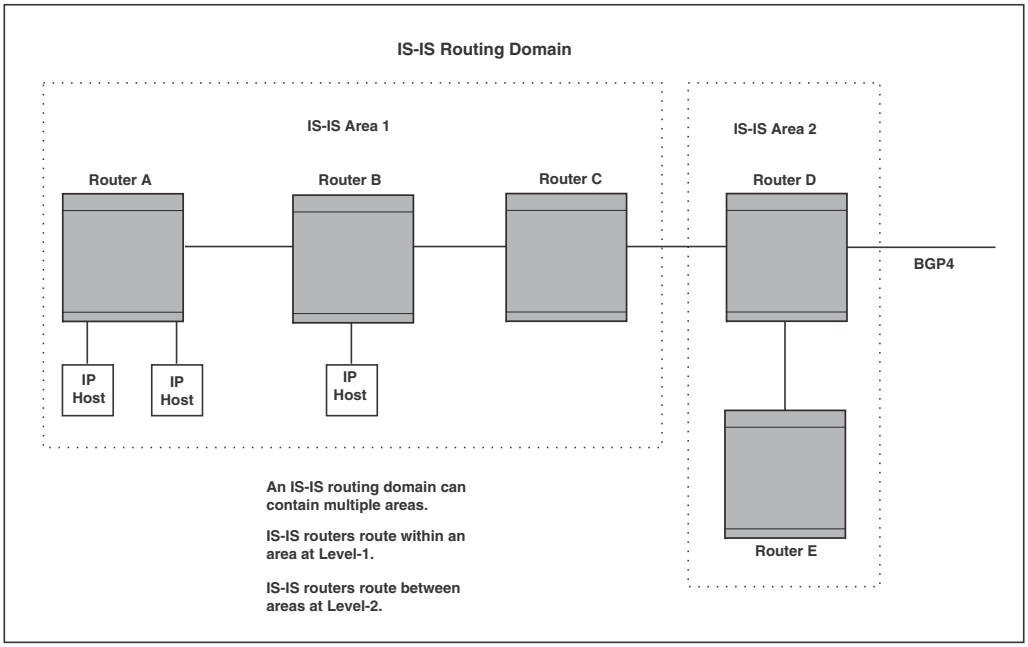
IS-IS uses the following categories to describe devices within an IS-IS routing domain (similar to an OSPF Autonomous System):

- ***Intermediate System (IS)*** – A device capable of forwarding packets from one device to another within the domain. In Internet Protocol (IP) terminology, an IS is a router.
- ***End System (ES)*** – A device capable of generating or receiving packets within the domain. In IP terminology, an ES is an end node or IP host.

When you configure IS-IS on a PowerConnect, the device is an IS.

Figure 141 shows an example of an IS-IS network.

FIGURE 141 An IS-IS network contains Intermediate Systems (ISs) and host systems



NOTE
Since the implementation of IS-IS does not route OSI traffic but instead routes IP traffic, IP hosts are shown instead of ESs.

The other basic IS-IS concepts illustrated in this figure are explained in the following sections.

Domain and areas

IS-IS is an IGP, and thus applies only to routes within a single routing domain. However, you can configure multiple areas within a domain. A PowerConnect can be a member of one area for each Network Entity Title (NET) you configure on the PowerConnect. The NET contains the area ID for the area the NET is in.

In Figure 141, Routers A, B, and C are in area 1. Routers D and E are in area 2. All the routers are in the same domain.

Level-1 routing and Level-2 routing

You can configure an IS-IS router such as a PowerConnect to perform one or both of the following levels of IS-IS routing[1]:

1. The ISO/IEC specifications use the spelling “routeing”, but this document uses the spelling “routing” to remain consistent with other Dell documentation.

- **Level-1** – A Level-1 router routes traffic only within the area the router is in. To forward traffic to another area, the Level-1 router sends the traffic to its nearest Level-2 router.
- **Level-2** – A Level-2 router routes traffic between areas within a domain.

In Figure 141 on page 935, Routers A and B are Level-1s only. Routers C and D are Level-1 and Level-2 ISs. Router E is a Level-1 IS only.

Neighbors and adjacencies

A PowerConnect configured for IS-IS forms an ***adjacency*** with each of the IS-IS devices to which it is directly connected. An adjacency is a two-way direct link (a link without router hops) over which the two devices can exchange IS-IS routes and other protocol-related information. The link is sometimes called a "circuit". The devices with which the PowerConnect forms adjacencies are its ***neighbors***, which are other ISs.

In Figure 141 on page 935, Router A has an IS-IS adjacency with Router B. Likewise, Router B has an IS-IS adjacency with Router A and Router C.

Designated IS

A ***Designated IS*** is an IS-IS router that is responsible for gathering and distributing link state information to other Level-1 or Level-2 ISs within the same broadcast network (LAN). The Level-1 and Level-2 Designated ISs within a broadcast network are independent, although the same PowerConnect can be a Level-1 Designated IS and a Level-2 Designated IS at the same time.

The Designated IS is elected based on the priority of each IS in the broadcast network. When an IS becomes operational, it sends a Level-1 or Level-2 Hello PDU to advertise itself to other ISs. If the IS is configured to be both a Level-1 and a Level-2 IS, the IS sends a separate advertisement for each level:

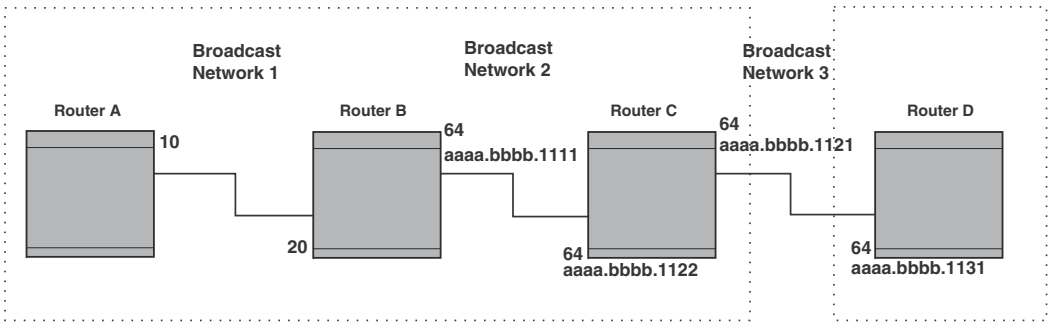
- The Level-1 IS that has the highest priority becomes the Level-1 Designated IS for the broadcast network.
- The Level-2 IS that has the highest priority becomes the Level-2 Designated IS for the broadcast network.

If the Designated IS becomes unavailable (for example, is rebooted), the IS with the next highest priority becomes the new IS. If two or more ISs have the highest priority, the IS with the highest MAC address becomes the Designated IS.

The priority is an interface parameter. Each interface that is enabled for IS-IS can have a different priority.

Figure 142 shows an example of the results of Designated IS elections. For simplicity, this example shows four of the five routers in Figure 141 on page 935, with the same domain and areas.

FIGURE 142 Each broadcast network has a Level-1 Designated IS and a Level-2 Designated IS



Designated IS election has the following results in this network topology:

- Router B is the Level-1 Designated IS for broadcast network 1
- Router C is the Level-1 Designated IS for broadcast network 2
- Router D is the Level-2 Designated IS for broadcast network 3

In this example, the IS-IS priorities for the IS-IS interfaces in broadcast network 1 have been changed by an administrator. The priorities for the interfaces in the other broadcast networks are still set to the default (64). When there is a tie, IS-IS selects the interface with the highest MAC address.

Broadcast pseudonode

In a broadcast network, the Designated IS maintains and distributes link state information to other ISs by maintaining a ***pseudonode***. A pseudonode is a logical host representing all the Level-1 or Level-2 links among the ISs in a broadcast network. Level-1 and Level-2 have separate pseudonodes, although the same device can be the pseudonode for Level-1 and Level-2.

Route calculation and selection

The Designated IS uses a ***Shortest Path First (SPF)*** algorithm to calculate paths to destination ISs and ESs. The SPF algorithm uses Link State PDUs (LSPDUs) received from other ISs as input, and creates the paths as output.

After calculating the paths, the Designated IS then selects the best paths and places them in the IS-IS route table. The Designated IS uses the following process to select the best paths.

1. Prefer the Level-1 path over the Level-2 path.
2. If there is no Level-1 path, prefer the internal Level-2 path over the external Level-2 path.
3. If there is still more than one path, prefer the path with the lowest metric.
4. If there is more than one path with the lowest metric, load share among the paths.

After selecting the best path to a destination, the software places the path in the IS-IS route table.

Three-way handshake for point-to-point adjacencies

Support was provided for Three-Way Handshake for Point-to-Point adjacencies as described in RFC 3373. This feature provides three-way handshake mechanisms on point-to-point interfaces for the following benefits:

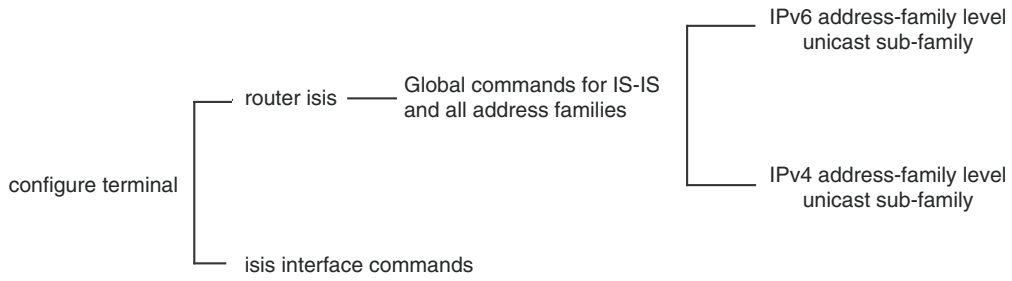
- Identifies neighbor restarts within the holding time period
- Identifies uni-directional link failures and stops forming of an adjacency with a peer where such link failures occur.

NOTE
This feature is the default operation and cannot be turned off. Routers with this feature are fully backward compatible with routers running an earlier release.

IS-IS CLI levels

The CLI includes various levels of commands for IS-IS. Figure 143 diagrams these levels.

FIGURE 143 IS-IS CLI levels



The IS-IS CLI levels are as follows:

- A global level for the configuration of the IS-IS protocol. At this level, all IS-IS configurations at this level apply to IPv4 and IPv6. You enter this layer using the **router isis** command.
  - Under the global level, you specify an address family. Address families to separate the IS-IS configurations for IPv4 and IPv6. You enter configurations that are for a specific You enter this level by entering the **address-family** command at the router isis level.
  - Under the address family level, you select a sub-address family, which is the type of routes for the configuration. For IS-IS, you specify **unicast**.
- An interface level.

Global configuration level

You enter the global configuration level of ISIS by entering the following command.

```
NetIron(config)#router isis
NetIron(config-isis-router)#
```

Syntax: [no] router isis

The (config-isis-router)# prompt indicates that you are at the global level for IS-IS. A configuration that you enter at this level applies to both IS-IS IPv4 and IS-IS IPv6.

Address family configuration level

The PowerConnect implementation of IS-IS includes the address family configuration level. Address families allow you to configure IPv4 IS-IS unicast settings that are separate and distinct from IPv6 IS-IS unicast settings (when IPv6 is supported).

Under the address family level, Dell currently supports the unicast address family configuration level only. The PowerConnect enters the IPv4 IS-IS unicast address family configuration level when you enter the following command while at the global IS-IS configuration level.

```
NetIron(config-isis-router)# address-family ipv4 unicast
NetIron(config-isis-router-ipv4u)#
```

Syntax: address-family ipv4 unicast

The (config-isis-router-ipv4u)# prompt indicates that you are at the IPv4 IS-IS unicast address family configuration level. While at this level, you can access several commands that allow you to configure IPv4 IS-IS unicast settings.

NOTE
Each address family configuration level allows you to access commands that apply to that particular address family only. To enable a feature in a particular address family, you must specify any associated commands for that feature in that particular address family. You cannot expect the feature, which you may have configured in the IPv4 IS-IS unicast address family, to work in the IPv6 IS-IS unicast address family unless it is explicitly configured in the IPv6 IS-IS unicast address family.

To exit from the ipv4 IS-IS unicast address family configuration level, enter the following command.

```
NetIron(config-isis-router-ipv4u)# exit-address-family
NetIron(config-isis-router)#
```

Entering this command returns you to the global IS-IS configuration level.

Interface level

Some IS-IS definitions are entered at the interface level. To enable IS-IS at the interface level, enter the following command.

```
NetIron(config)# interface ethernet 2/3
NetIron(config-if-e1000-2/3)#ip router isis
```

Syntax: [no] ip router isis

Enabling IS-IS globally

To configure IPv4 IS-IS, perform the tasks listed below.

1. Globally enable IS-IS by entering the following command.

   ```
   NetIron(config)# router isis
   ISIS: Please configure NET!
   ```

   Once you enter **router isis**, the device enters the IS-IS router configuration level.

Syntax: [no] router isis

To disable IS-IS, use the **no** form of this command.

2. If you have not already configured a NET for IS-IS, enter commands such as the following.

```
NetIron(config-isis-router)# net 49.2211.aaaa.bbbb.cccc.00
NetIron(config-isis-router)#
```

The commands in the example above configure a NET that has the area ID 49.2211, the system ID aaaa.bbbb.cccc (the device's base MAC address), and SEL value 00.

Syntax: [no] net <*area-id*>.<*system-id*>.<*sel*>

The <*area-id*> parameter specifies the area and has the format xx or xx.xxxx. For example, 49 and 49.2211 are valid area IDs.

The <*system-id*> parameter specifies the router's unique IS-IS router ID and has the format xxxx.xxxx.xxxx. You can specify any value for the system ID. A common practice is to use the device's base MAC address as the system ID. The base MAC address is also the MAC address of port 1. To determine the base MAC address, enter the following command at any level of the CLI: **show interfaces brief**. The base MAC address is listed in the first row of information, in the MAC column.

You must use the same system ID in all the NETs on the PowerConnect

NOTE
The parameter descriptions above are the recommended values for the NET. However, the CLI accepts any value that fits within the following lengths and formats:

xx.xxxx.xxxx.xxxx.00 – minimum length of NET

xx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.00 – maximum length of NET

The <*sel*> parameter specifies the NSAP Selector (SEL). This value must always be 00 (two zeros). The value 00 indicates that this address is an NET.

To delete a NET, use the **no** form of this command.

3. Configure ISIS parameters. Refer to the sections "Globally configuring IS-IS on a device" on page 940, "Configuring IPv4 address family route parameters" on page 951, and "Configuring ISIS properties on an interface" on page 968.

None of the IS-IS parameters require a software reload to places changes into effect and most parameter changes take effect immediately. However, changes for the following parameters take effect only after you disable and then re-enable redistribution:

- Change the default metric.
- Add, change, or negate route redistribution parameters.

Some IS-IS parameter changes take effect immediately while others do not take full effect until you disable, then re-enable route redistribution.

Globally configuring IS-IS on a device

This section describes how to change the global IS-IS parameters. These parameter settings apply to both IS-IS IPv4 and IS-IS IPv6.

Setting the overload bit

If an IS's resources are overloaded and are preventing the IS from properly performing IS-IS routing, the IS can inform other ISs of this condition by setting the overload bit in LSPDUs sent to other ISs from 0 (off) to 1 (on).

When an IS is overloaded, other ISs will not use the overloaded IS to forward traffic. An IS can be in the overload state for Level-1, Level-2, or both as described in the following section:

- If an IS is in the overload state for Level-1, other Level-1 ISs stop using the overloaded IS to forward Level-1 traffic. However, the IS can still forward Level-2 traffic, if applicable.
- If an IS is in the overload state for Level-2, other Level-2 ISs stop using the overloaded IS to forward Level-2 traffic. However, the IS can still forward Level-1 traffic, if applicable.
- If an IS is in the overload state for both levels, the IS cannot forward traffic at either level.

By default, the PowerConnect automatically sets the overload bit to 1 (on) in its LSPDUs to other ISs if an overload condition occurs.

You can set the overload bit on to administratively shut down IS-IS without disabling the protocol. Setting the overload bit on is useful when you want to make configuration changes without removing the PowerConnect from the network.

In addition, you can configure the PowerConnect to set the overload bit on for a specific number of seconds during startup, to allow IS-IS to become fully active before the device begins IS-IS routing. By default, there is no delay (0 seconds).

To immediately set the overload bit on, enter the following command.

```
NetIron(config-isis-router)# set-overload-bit
```

This command administratively shuts down IS-IS by configuring the PowerConnect to immediately set the overload bit to 1 (on) in all LSPs sent to other ISs.

To configure the PowerConnect to temporarily set the overload bit on after a software reload, enter a command such as the following.

```
NetIron(config-isis-router)# set-overload-bit on-startup 5
```

This command configures the PowerConnect to set the overload bit on in all its IS-IS LSPs sent to other ISs during the first five seconds following a successful software reload. After the five seconds expire, the PowerConnect resets the overload bit to off in all its IS-IS LSPs.

Syntax: [no] set-overload-bit [on-startup <*secs*>**]**

The **on-startup** <*secs*> parameter specifies the number of seconds following a reload to set the overload bit on. You can specify a number from 5 – 86400 (24 hours).

A new option has been added to the **set-overload-bit** command to prevent route black holing in support of RFC 3277. With this option set, the behavior of ISIS will be changed during a router reboot. During a router reboot, ISIS sets the overload bit in its LSPDUs until BGP has converged.

This feature is configured using the **set-overload-bit** command as shown in the following.

```
NetIron(config-isis-router)# set-overload-bit on-startup wait-for-bgp 1000
```

Syntax: [no] set-overload-bit on-startup wait-for-bgp <*max-bgp-wait-time*>

The <*max-bgp-wait-time*> variable is the maximum time IS-IS will wait for BGP convergence to complete. Once this time has been exceeded without BGP converging, IS-IS will exit the overload state. The default value is 600 seconds (10 minutes), possible values range: 5 to 86400 seconds.

Configuring authentication

By default, a PowerConnect router does not authenticate packets sent to or received from an end system (ES) or other intermediate system (IS). In previous releases, the Multi-Service IronWare software let you configure area, domain, and circuit passwords to direct the PowerConnect router to check for a password in packets sent from the device.

The new method of configuring an authentication password introduces the option of using the Hashed Message Authentication codes - Message Digest 5 (HMAC-MD5) algorithm.

This implementation is in conformance with RFC 3567 - Intermediate System to Intermediate System (IS-IS) Cryptographic Authentication.

NOTE
The commands for setting the password used in previous versions of the Multi-Service IronWare software are now hidden in the CLI, however they are backward compatible and will operate in this release.

Configuring ISIS authentication at the Router ISIS mode

To configure ISIS authentication at the Router ISIS mode on a PowerConnect router, you must perform the following tasks:

- Configure ISIS Authentication Mode
- Configure ISIS Authentication Key
- Disable ISIS Authentication Check (optional)

Configuring ISIS authentication mode

The following commands configure the ISIS for the authentication mode.

```
NetIron(config)# router isis
NetIron(config-isis-router)# auth-mode md5 level-1
```

Syntax: [no] auth-mode [cleartext | md5] [level-1 | level-2]

The **cleartext** parameter specifies that the ISIS PDUs will be authenticated using a cleartext password.

The **md5** parameter specifies that the ISIS PDUs will be authenticated using the Hashed Message Authentication codes - Message Digest 5 (HMAC-MD5) algorithm.

The **level-1** parameter specifies that the authentication type-length-value (TLV) tuple be added to the L1 LSP, L1 CSNP, and LI PSNP packets.

The **level-2** parameter specifies that the authentication TLV tuple be added to the L2 LSP, L2 CSNP, and L2 PSNP packets.

Configuring ISIS authentication key

The following commands configure an authentication key to be used with the mode specified in "Configuring ISIS authentication mode".

```
NetIron(config)# router isis
NetIron(config-isis-router)# auth-mode md5 level-1
NetIron(config-isis-router)# auth-key supervisor level-1
NetIron(config-isis-router)# auth-key supervisor level-2
```

Syntax: [no] auth-key <*string*> **[level-1 | level-2]**

The <*string*> variable specifies a text string that is used as an authentication password. The authentication mode must be configured before this value can be configured.

By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **auth-key** and <*string*>.

Example

```
NetIron(config-isis-router)# auth-key 0 supervisor level-1
```

The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code "2".

```
auth-key 2 $on-n level-1
```

The prefix can be one of the following:

- 0 = the key string is not encrypted and is in clear text
- 1 = the key string uses proprietary simple crytographic 2-way algorithm
- 2 = the key string uses proprietary base64 crytographic 2-way algorithm

The **level-1** parameter specifies that the authentication key specified here is used to authenticate the L1 LSP, L1 CSNP and LI PSNP packets.

The **level-2** parameter specifies that the authentication key specified here is used to authenticate the L2 LSP, L2 CSNP and L2 PSNP packets.

You must enter a configuration for both level-1 and level-2 in order to enter the auth-key string.

NOTE
If the authentication mode is reset for the level specified, the authentication key must also be reset.

Disabling ISIS authentication checking

When transitioning from one authentication mode to another, changing the authentication mode can cause packets to drop because only some of the routers have been reconfigured. During such a transition, it can be useful to disable ISIS authentication checking temporarily until all routers are reconfigured and the network is stable.

You can use the following commands to disable ISIS authentication checking.

```
NetIron(config)# router isis
NetIron(config-isis-router)# no auth-check level-1
```

Syntax: [no] auth-check [level-1 | level-2]

This command enables and disables ISIS authentication checking. The default is enabled and the **[no]** parameter disables authentication checking.

The **level-1** parameter specifies that authentication checking is enabled/ disabled for L1 LSP, L1 CSNP and LI PSNP packets.

The **level-2** parameter specifies that authentication checking is enabled/disabled for L2 LSP, L2 CSNP and L2 PSNP packets.

Configuring ISIS MD5 authentication on a specified interface

To configure ISIS MD5 authentication on a specified interface on a PowerConnect router, you must perform the following tasks:

- Configure ISIS Interface Authentication Mode for a Specified Interface
- Configure ISIS Authentication Key on the Interface
- Disable ISIS Authentication Check on an Interface (optional)

Configuring ISIS authentication mode for a specified interface

The following commands configure the ISIS for the authentication mode on a specified interface.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e10000-3/1)# isis auth-mode md5 level-1
```

Syntax: [no] isis auth-mode [cleartext | md5] [level-1 | level-2]

The **cleartext** parameter specifies that the ISIS PDUs will be authenticated using a cleartext password.

The **md5** parameter specifies that the ISIS PDUs authenticated using the Hashed Message Authentication codes - Message Digest 5 (HMAC-MD5) algorithm.

The **level-1** parameter specifies that the authentication TLV tuple be added to the L1 Hello packets.

The **level-2** parameter specifies that the authentication TLV tuple be added to the L2 Hello packets.

NOTE
If either level-1 or level-2 are not specified, the configuration is applied to both level-1 and level-2.

Configuring an ISIS authentication key for a specified interface

The following commands configure an authentication key to be used with the mode specified in "Configuring ISIS authentication mode for a specified interface".

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e10000-3/1)# isis auth-key supervisor level-1
```

Syntax: [no] isis auth-key <*key*> **[level-1 | level-2]**

The <*key*> value specifies a text string that is used as an authentication password. The authentication mode must be configured before this value can be configured.

The **level-1** parameter specifies that the authentication key specified here is used to authenticate the L1 Hello packets.

The **level-2** parameter specifies that the authentication key specified here is used to authenticate the L2 Hello packets.

NOTE
If either level-1 or level-2 are not specified, the configuration is applied to both level-1 and level-2.

NOTE
If the authentication mode is reset for the level specified, the authentication key must also be reset.

NOTE
The **isis auth-key** command allows the user to configure more 80 characters, but only the first 80 characters are used.

Disabling ISIS authentication checking on a specified interface

When transitioning from one authentication mode to another, changing the authentication mode can cause packets to drop because only some of the routers have been reconfigured. During such a transition, it can be useful to disable ISIS authentication checking temporarily until all routers are reconfigured and the network is stable.

You can use the following commands to disable ISIS authentication checking on a specified interface.

```
NetIron(config)# interface ethernet 3/1
NetIron(if-e10000-3/1)# no isis auth-check level-1
```

Syntax: [no] isis auth-check [level-1 | level-2]

This command enables and disables ISIS authentication checking. The default is enabled and the **[no]** parameter disables authentication checking.

The **level-1** parameter specifies that authentication checking is enabled/ disabled for L1 Hello packets.

The **level-2** parameter specifies that authentication checking is enabled/disabled for L2 Hello packets.

NOTE
If either level-1 or level-2 are not specified, the configuration is applied to both level-1 and level-2.

Changing the IS-IS level globally

By default, a PowerConnect can operate as both a Level-1 and IS-IS Level-2 router. To globally change the level supported from Level-1 and Level-2 to Level-1 only, enter the following command.

```
NetIron(config-isis-router)# is-type level-1
```

Syntax: [no] is-type level-1 | level-1-2 | level-2

The **level-1 | level-1-2 | level-2** parameter specifies the IS-IS type. If you want to re-enable support for both IS-IS types, re-enter the command you entered to change the IS-IS type, and use “no” in front of the command.

To change the IS-IS on an interface, refer to “Changing the IS-IS level on an interface” on page 970.

Disabling or re-enabling display of hostname

Dell’s implementation of IS-IS supports RFC 2763, which describes a mechanism for mapping IS-IS system IDs to the hostnames of the devices with those IDs. For example, if you set the hostname on the PowerConnect to “IS-IS Router 1”, the mapping feature uses this name instead of the PowerConnect’s IS-IS system ID in the output of the following commands:

- **show isis database**
- **show isis interface**
- **show isis neighbor**

The PowerConnect's hostname is displayed in each CLI command prompt, for example.

```
NetIron(config-isis-router)#
```

The name mapping feature is enabled by default. If you want to disable name mapping, enter the following command.

```
NetIron(config-isis-router)# no hostname
```

Syntax: [no] hostname

To display the name mappings, enter the **show isis hostname** command.

Changing the Sequence Numbers PDU interval

A ***Complete Sequence Numbers PDU (CSNP)*** is a complete list of the LSPs in the Designated IS' link state database. The CSNP contains a list of all the LSPs in the database, as well as other information that helps IS neighbors determine whether their LSP databases are in sync with one another. The Designated IS sends CSNPs to the broadcast interface. Level-1 and Level-2 each have their own Designated IS.

A ***Partial Sequence Numbers PDU (PSNP)*** is a partial list of LSPs. ISs other than the Designated IS (that is, the non-Designated ISs) send PSNPs to the broadcast interface.

The CSNP interval specifies how often the Designated IS sends a CSNP to the broadcast interface. Likewise, the PSNP interval specifies how often other ISs (non-Designated ISs) send a PSNP to the broadcast interface.

The interval you can configure on the PowerConnect applies to both Level-1 and Level-2 CSNPs and PSNPs. The default interval is 10 seconds. You can set the interval to a value from 0 – 65535 seconds.

To change the interval, enter a command such as the following.

```
NetIron(config-isis-router)# csnp-interval 15
```

Syntax: [no] csnp-interval <*secs*>

The <*secs*> parameter specifies the interval and can be from 0 – 65535 seconds. The default is 10 seconds.

NOTE
PSNP has a default interval of 2 seconds and is not configurable.

Changing the maximum LSP lifetime

The maximum LSP lifetime is the maximum number of seconds an unrefreshed LSP can remain in the PowerConnect's LSP database. The maximum LSP lifetime can be from 1 – 65535 seconds. The default is 1200 seconds (20 minutes).

To change the maximum LSP lifetime to 2400 seconds, enter a command such as the following.

```
NetIron(config-isis-router)# max-lsp-lifetime 2400
```

Syntax: [no] max-lsp-lifetime <*secs*>

The <*secs*> parameter specifies the maximum LSP lifetime and can be from 1 – 65535 seconds. The default is 1200 seconds (20 minutes).

NOTE
The **max-lsp-lifetime** and the **lsp-refresh-interval** must be set in such a way that the LSPs are refreshed before the **max-lsp-lifetime** expires; otherwise, the PowerConnect's originated LSPs may be timed out by it's neighbors. Refer to "Changing the LSP refresh interval" on page 947.

Changing the LSP refresh interval

The LSP refresh interval is the maximum number of seconds the PowerConnect waits between sending updated LSPs to its IS-IS neighbors. The interval can be from 1 – 65535 seconds. The default is 900 seconds.

To change the LSP refresh interval to 20000 seconds, enter a command such as the following.

```
NetIron(config-isis-router)# lsp-refresh-interval 20000
```

Syntax: [no] lsp-refresh-interval <*secs*>

The <*secs*> parameter specifies the maximum refresh interval and can be from 1 – 65535 seconds. The default is 900 seconds (15 minutes).

Changing the LSP generation interval

The LSP generation interval is the minimum number of seconds the PowerConnect waits between sending updated LSPs to its IS-IS neighbors. The interval can be from 1 – 120 seconds. The default is 10 seconds.

To change the LSP generation interval to 45 seconds, enter a command such as the following.

```
NetIron(config-isis-router)# lsp-gen-interval 45
```

Syntax: [no] lsp-gen-interval <*secs*>

The <*secs*> parameter specifies the minimum refresh interval and can be from 1 – 120 seconds. The default is 10 seconds.

Changing the LSP interval and retransmit interval

You LSP interval is the rate of transmission, in milliseconds of the LSPs. The retransmit interval is the time the device waits before it retransmits LSPs. To define an LSP interval, enter a command such as the following.

```
NetIron(config-isis-router)# lsp-interval 45
```

Syntax: [no] lsp-interval <*milliseconds*>

Enter 1 – 4294967295 milliseconds for the LSP interval. The default is 33 milliseconds.

To define an interval for retransmission of LSPs enter a command such as the following.

```
NetIron(config-isis-router)# retransmit-interval 3
```

Syntax: [no] retransmit-interval <*seconds*>

Enter 0 – 65535 seconds for the retransmission interval. The default is 5 seconds.

Changing the SPF timer

Every IS maintains a Shortest Path First (SPF) tree, which is a representation of the states of each of the IS's links to ESs and other ISs. If the IS is both a Level-1 and Level-2 IS, it maintains separate SPF trees for each level.

To ensure that the SPF tree remains current, the IS updates the tree at regular intervals following a change in network topology or the link state database. By default, the PowerConnect recalculates its IS-IS tree every five seconds following a change. You can change the SPF timer to a value from 1 – 120 seconds.

NOTE
This command has been superseded by the IS-IS PSPF Exponential back-off feature.

To change the SPF interval, enter a command such as the following.

```
NetIron(config-isis-router)# spf-interval 30
```

Syntax: [no] spf-interval <*secs*>

The <*secs*> parameter specifies the interval and can be from 1 – 120 seconds. The default is 5 seconds.

Configuring the IS-IS PSPF exponential back-off feature

The PowerConnect router uses the exponential back-off mechanism to provide a more responsive approach to running the PSPF calculations. With this new feature, there is a new configurable command called **partial-spf-interval** that allows you to schedule PSPF processing as described in the following.

An **initial-wait** interval can be configured as a wait time after an LSP change until the first PSPF calculation. Optionally, this value is followed by another configurable variable called the **second-wait** interval that is used as a wait time between the first and second PSPF calculations. The **second-wait** interval (if configured) is then increased in multiples of 2 until it reaches the maximum hold time as configured by the **max-wait** variable. Once reached, the maximum hold time remains the hold interval between PSPF calculations until there are no further changes in the network. When there are no network changes in a hold down period, the gap between PSPF calculations returns to the **initial-wait** interval and the process begins again. '

If an **initial-wait** interval is configured without a **second-wait** interval, the **max-wait** variable is used for the second and all subsequent intervals.

If the **initial-wait** and **second-wait** intervals are not configured, the **max-wait** variable is used for the first and all subsequent intervals.

The IS-IS PSPF exponential back-off mechanism is configured using the **partial-spf-interval** command , as shown in the following.

```
NetIron(config-isis-router)# partial-spf-interval 60 1000 5000
```

Syntax: [no] partial-spf-interval <*max-wait*> <*initial-wait*> <*second-wait*>

The <*max-wait*> variable specifies the maximum interval between PSPF recalculations. The range of acceptable values is 0 – 120000 milliseconds. The default is 5000 milliseconds (5 seconds).

The <*initial-wait*> variable is an optional value that specifies the wait time after an LSP change until the first PSPF calculation. The range of acceptable values is 0 – 120000 milliseconds. The default for this variable is value of the **max-wait** time.

The <*second-wait*> variable is an optional value that specifies the wait time between the first and second PSPF calculations. If this optional value is configured, it will be doubled with each PSPF recalculation until the value is equal to the <*spf-max-wait*> value. The range of acceptable values is 0 – 120000 milliseconds. The default for this variable is value of the **max-wait** time.

Configuring the IS-IS flooding mechanism

The IS-IS fast flooding feature allows you to configure IS-IS on the router to flood Link State PDUs to other routers in the network before running SPF. This improves database synchronization by allowing LSP changes to be propagated to neighbors before running SPF. The IS-IS fast-flood feature is implemented using the fast-flood command as shown in the following.

```
NetIron(config-isis-router)# fast-flood 10
```

Syntax: [no] fast-flood <*lsp-count*>

The <*lsp-count*> variable sets the number of LSPs that trigger SPF that must be flooded before running SPF. The SPF run will be delayed until the configured number of LSPs have been flooded. If the number of changed LSPs is less than the configured number, then only the changed LSPs are flooded. The variable can be set to the following values: 1 - 25. This variable is optional and will be set to a value of 4 if not specified.

Globally disabling or re-enabling hello padding

By default, the PowerConnect adds extra data to the end of a hello packet to make the packet the same size as the maximum length of PDU the PowerConnect supports.

The padding applies to the following types of hello packets:

- ES hello (ESH PDU)
- IS hello (ISH PDU)
- IS to IS hello (IIH PDU)

The padding consists of arbitrarily valued octets. A padded hello PDU indicates the largest PDU that the PowerConnect can receive. Other ISs that receive a padded hello PDU from the PowerConnect can therefore ensure that the IS-IS PDUs they send the PowerConnect. Similarly, if the PowerConnect receives a padded hello PDU from a neighbor IS, the PowerConnect knows the maximum size PDU that the PowerConnect can send to the neighbor.

When padding is enabled, the maximum length of a Hello PDU sent by the PowerConnect is 1514 bytes.

If you need to disable padding, you can do so globally or on individual interfaces. Generally, you do not need to disable padding unless a link is experiencing slow performance. If you enable or disable padding on an interface, the interface setting overrides the global setting.

To globally disable padding of IS-IS hello PDUs, enter the following command.

```
NetIron(config-isis-router)# no hello padding
```

This command disables all hello PDU padding on the PowerConnect. To re-enable padding, enter the following command.

```
NetIron(config-isis-router)# hello padding
```

Syntax: [no] hello padding [point-to-point]

By default, hello padding is enabled. Enter the **no** form of the command to disable hello padding.

The **point-to-point** option enables hello PDU padding on Point-to-Point interfaces.

To disable hello padding on an interface, refer to "Disabling and enabling hello padding on an interface" on page 970.

Logging adjacency changes

The PowerConnect can be configured to log changes in the status of an adjacency with another IS. Logging of the adjacency changes is disabled by default. To enable or disable them, use either of the following methods.

To enable logging of adjacency changes, enter the following command.

```
NetIron(config-isis-router)# log adjacency
```

Syntax: [no] log adjacency

To disable logging of adjacency changes, enter the following command.

```
NetIron(config-isis-router)# no log adjacency
```

Logging invalid LSP packets received

The PowerConnect can be configured to provide logging of invalid LSP packets. Logging of the invalid LSP packets is disabled by default. To enable or disable this function, use either of the following methods.

To enable logging of invalid LSP packets, enter the following command.

```
NetIron(config-isis-router)# log invalid-lsp-packets
```

Syntax: [no] log invalid-lsp-packets

To disable logging of invalid LSP packets, enter the following command.

```
NetIron(config-isis-router)# no log invalid-lsp-packets
```

Disabling partial SPF optimizations

IS-IS employs certain partial SPF optimizations to make partial changes to the routing table in network change situations where the topology of the network has not changed but where there may be changes in the IP networks advertised by routers. These optimizations are termed partial SPF optimizations.

You can optionally configure IS-IS to perform a full SPF calculation when any network (non-topology) change occurs by using the **disable-partial-spf-opt** command. When **disable-partial-spf-opt** is configured, IS-IS always runs full SPF for all such network changes.

To disable partial SPF calculations for IS-IS, enter the following command.

```
NetIron(config-isis-router)# disable-partial-spf-opt
```

Syntax: [no] disable-partial-spf-opt

To restore partial SPF optimizations, use the **no** form of this command.

Disabling incremental SPF optimizations

In the event of certain topology changes (for instance non-local adjacency flaps), IS-IS employs incremental SPF optimizations to efficiently update the routing table. An incremental SPF is faster and takes fewer CPU cycles than a full SPF.

You can optionally configure IS-IS to perform a full SPF calculation when any network topology change occurs by using the **disable-incremental-spf-opt** command. When **disable-incremental-spf-opt** is configured, IS-IS always runs full SPF for all such network topology changes.

To disable incremental SPF optimizations for IS-IS, enter the following command.

```
NetIron(config-isis-router)# disable-incremental-spf-opt
```

Syntax: [no] disable-incremental-spf-opt

To restore incremental SPF optimizations, use the **no** form of this command.

NOTE
If you disable the partial SPF optimizations (by using the **disable-partial-spf-opt** command), IS-IS automatically disables the incremental SPF optimizations and always runs full SPF, too. However, the reverse is not true: disabling incremental SPF optimizations does not disable partial optimizations.

Configuring IPv4 address family route parameters

This section describes how to modify the IS-IS parameters for the IS-IS IPv4 unicast address family. To enter the IPv4 unicast address family, refer to the "Address family configuration level" on page 939.

Changing the metric style

The metric style specifies the Types, Lengths, and Values (TLVs) an IS-IS LSP can have. The TLVs specify the types of data, the maximum length of the data, and the valid values for the data. One of the types of data the TLVs control is a route's default-metric. By default, the PowerConnect uses the standard IS-IS TLVs, which allows metric values from 1 – 63. The default metric style is called "narrow." You can increase the range of metric values supported by the PowerConnect by changing the metric style to wide. The wide metric style allows metric values in the range 1 – 16777215.

To change the metric style to wide, enter the following command.

```
NetIron(config-isis-router-ipv4)# metric-style wide
```

This command changes the metric style for both Level-1 and Level-2.

Syntax: [no] metric-style wide [level-1 | level-2]

The **level-1 | level-2** parameter specifies the levels to which the change applies. If not specified, the changes are applied to both levels.

Changing the maximum number of load sharing paths

By default, IPv4 IS-IS can calculate and install four equal-cost paths into the IPv4 forwarding table. You can change the number of paths IPv4 IS-IS can calculate and install in the IPv4 forwarding table to a value from
1 – 8. If you change the number of paths to one, the PowerConnect does not load share multiple route paths learned from IPv4 IS-IS.

For example, to change the number of paths IPv4 IS-IS can calculate and install in the IPv4 forwarding table to three, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# maximum-paths 4
```

Syntax: [no] maximum-paths <*number*>

The <*number*> parameter specifies the number of paths IPv4 IS-IS can calculate and install in the IPv4 forwarding table. Enter a number from 1 to 4. The value specified in <*number*> is limited by the ip load-sharing value.

To return to the default number of maximum paths, enter the **no** form of this command.

Enabling advertisement of a default route

By default, the PowerConnect does not generate or advertise a default route to its neighboring ISs. A default route is not advertised even if the device's IPv4 route table contains a default route. You can enable the device to advertise a default route to all neighboring ISs using one of the following methods. By default, the feature originates the default route at Level 2 only. However, you can apply a route map to originate the default route to Level 1 only or at both Level 1 and Level 2.

NOTE
This feature requires the presence of a default route in the IPv4 route table.

To enable the PowerConnect to advertise a default route that is originated a Level 2, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# default-information-originate
```

This command enables the device to advertise a default route into the IPv4 IS-IS area to which the device is attached.

Syntax: [no] default-information-originate [route-map <*name*>**]**

The **route-map** <*name*> parameter allows you to specify the level on which to advertise the default route. You can specify one of the following:

- Advertise to Level-1 ISs only.
- Advertise to Level-2 ISs only.
- Advertise to Level-1 and Level-2 ISs.

NOTE
The route map must be configured before you can use the route map as a parameter with the **default-information-originate** command.

To use a route map to specify the router to advertise a default route to Level 1, enter commands such as the following at the Global CONFIG level.

```
NetIron(config)# route-map default_level1 permit 1
NetIron(config-routemap default_level1)# set level level-1
NetIron(config-routemap default_level1)# exit
NetIron(config)# router isis
NetIron(config-isis-router)# address-family ipv4 unicast
NetIron(config-isis-router-ipv4u)# default-information-originate route-map
default_level1
```

These commands configure a route map to set the default advertisement level to Level 1 only.

Syntax: [no] route-map <*map-name*> **permit | deny** <*sequence-number*>

Syntax: [no] set level level-1 | level-1-2 | level-2

For this use of a route map, use the **permit** option and do not specify a **match** statement. Specify a **set** statement to set the level to one of the following:

- **level-1** – Level 1 only.
- **level-1-2** – Level 1 and Level 2.
- **level-2** – Level 2 only (default).

Matching based on ISIS protocol type

The **match** option has been added to the **route-map** command that allows IS-IS routes to be matched based on level-1 or level-2 or all IS-IS routes. .

```
NetIron(config-routemap test)# match protocol isis level-1
```

Syntax: [no] match protocol isis {level-1|level-2}

The **match protocol isis level-1** option can be used to match the IS-IS Level-1 routes.

The **match protocol isis level-2** option can be used to match the IS-IS Level-2 routes.

Changing the administrative distance for IPv4 IS-IS

When the PowerConnect has paths from multiple routing protocols to the same destination, it compares the administrative distances of the paths and selects the path with the lowest administrative distance to place in the IPv4 route table.

For example, if the router has a path from RIP, from OSPF, and IPv4 IS-IS to the same destination, and all the paths are using their protocols' default administrative distances, the router selects the OSPF path, because that path has a lower administrative distance than the RIP and IPv4 IS-IS paths.

Here are the default IPv4 administrative distances on the PowerConnect:

- Directly connected – 0 (this value is not configurable)
- Static – 1 (applies to all static routes, including default routes)
- EBGP – 20
- OSPF – 110
- IPv4 IS-IS – 115
- RIP – 120
- IBGP – 200
- Local BGP – 200

- Unknown – 255 (the device will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the PowerConnect receives routes for the same network from IPv4 IS-IS and from RIP, it will prefer the IPv4 IS-IS route by default.

To change the administrative distance for IPv4 IS-IS routes, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# distance 100
```

Syntax: [no] distance <*number*>

This command changes the administrative distance for all IPv4 IS-IS routes to 100.

The <*number*> parameter specifies the administrative distance. You can specify a value from 1 – 255. (Routes with a distance value of 255 are not installed in the routing table.) The default for IPv4 IS-IS is 115.

Configuring summary addresses

You can configure summary addresses to aggregate IS-IS route information. Summary addresses can enhance performance by reducing the size of the Link State database, reducing the amount of data the PowerConnect needs to send to its neighbors, and reducing the CPU cycles used for IS-IS.

When you configure a summary address, the address applies only to Level-2 routes by default. You can specify Level-1 only, Level-2 only, or Level-1 and Level-2 when you configure the address.

To configure a summary address, enter a command such as the following.

```
NetIron(config-isis-router-ipv4u)# summary-address 192.168.0.0 255.255.0.0
```

This command configures a summary address for all Level-2 IS-IS route destinations between 192.168.1.0 – 192.168.255.255.

Syntax: [no] summary-address <*ip-addr*> <*subnet-mask*> **[level-1 | level-1-2 | level-2]**

The <*ip-addr*> <*subnet-mask*> parameters specify the aggregate address. The mask indicates the significant bits in the address. Ones are significant, and zeros allow any value. In the command example above, the mask 255.255.0.0 matches on all addresses that begin with 192.168 and contain any values for the final two octets.

The **level-1 | level-1-2 | level-2** parameter specifies the route types to which the aggregate route applies. The default is **level-2**.

Redistributing routes into IPv4 IS-IS

To redistribute routes into IPv4 IS-IS, you can perform the following configuration tasks:

- Change the default redistribution metric (optional).
- Configure the redistribution of a particular route type into IPv4 IS-IS (mandatory).

The PowerConnect can redistribute routes from the following route sources into IPv4 IS-IS:

- BGP4+.
- RIP.
- OSPF.
- Static IPv4 routes.

- IPv4 routes learned from directly connected networks.

The PowerConnect can also can redistribute Level-1 IPv4 IS-IS routes into Level-2 IPv4 IS-IS routes, and Level-2 IPv4 IS-IS routes into Level-1 IPv4 IS-IS routes.

Route redistribution from other sources into IPv4 IS-IS is disabled by default. When you enable redistribution, the device redistributes routes only into Level 2 by default. You can specify Level 1 only, Level 2 only, or Level 1 and Level 2 when you enable redistribution.

The device automatically redistributes Level-1 routes into Level-2 routes. Thus, you do not need to enable this type of redistribution. You also can enable redistribution of Level-2 routes into Level-1 routes.

The device attempts to use the redistributed route's metric as the route's IPv4 IS-IS metric. For example, if an OSPF route has an OSPF cost of 20, the router uses 20 as the route's IPv4 IS-IS metric. The device uses the redistributed route's metric as the IPv4 IS-IS metric unless the route does not a have a valid metric. In this case, the device assigns the default metric value to the route. For information about the default metric, refer to the "Changing the default redistribution metric" section, which follows this section.

Changing the default redistribution metric

When IPv4 IS-IS redistributes a route from another route source (such as OSPF, BGP4+, or a static IPv4 route) into IPv4 IS-IS, it uses the route's metric value as its metric when the metric is not modified by a route map or metric parameter and the default redistribution metric is set to its default value of 0. You can change the default metric to a value from 0 – 65535.

NOTE
The implementation of IS-IS does not support the optional metric types Delay, Expense, or Error.

For example, to change the default metric to 20, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# default-metric 20
```

Syntax: [no] default-metric <*value*>

The <*value*> parameter specifies the default metric. You can specify a value from 0 – 65535. The default is 0.

To restore the default value for the default metric, enter the **no** form of this command.

Redistributing static IPv4 routes into IPv4 IS-IS

To redistribute static IPv4 routes from the IPv4 static route table into IPv4 IS-IS routes, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# redistribute static
```

This command configures the PowerConnect to redistribute all static IPv4 routes into Level-2 IS-IS routes.

Syntax: [no] redistribute static [level-1 | level-1-2 | level-2] |
metric <*num*> **| metric-type [external | internal] | route-map** <*name*>

The **level-1**, **level-1-2**, and **level-2** keywords restrict redistribution to the specified IPv4 IS-IS level.

The **metric** <*num*> parameter changes the metric. You can specify a value from 0 - 4294967295.

The **metric-type external** | **internal** parameter restricts redistribution to one of the following:

- **external** – The metric value is not comparable to an IPv4 IS-IS internal metric and is always higher than the IPv4 IS-IS internal metric.
- **internal** – The metric value is comparable to metric values used by IPv4 IS-IS. This is the default.

The **route-map** <*name*> parameter restricts redistribution to those routes that match the specified route map. The route map must already be configured before you use the route map name with the **redistribute** command. For example, to configure a route map that redistributes only the static IPv4 routes to the destination networks 192.168.0.0/24, enter commands such as the following:

```
NetIron(config)# access-list 10 permit 192.168.0.0 0.0.255.255
NetIron(config)# route-map static permit 1
NetIron(config-routemap static)# match ip address 10
NetIron(config-routemap static)# router isis
NetIron(config-isis-router)# address-family ipv4 unicast
NetIron(config-isis-router-ipv4u)# redistribute static route-map static
```

Redistributing directly connected routes into IPv4 IS-IS

To redistribute directly connected IPv4 routes into IPv4 IS-IS routes, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# redistribute connected
```

This command configures the PowerConnect to redistribute all directly connected routes in the IPv4 route table into Level-2 IPv4 IS-IS.

Syntax: [no] **redistribute connected [level-1 | level-1-2 | level-2] | metric** <*number*> **| metric-type [external | internal] | route-map** <*name*>]

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing RIP routes into IPv4 IS-IS

To redistribute RIP routes into IPv4 IS-IS, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# redistribute rip
```

This command configures the PowerConnect to redistribute all RIP routes into Level-2 IS-IS.

Syntax: [no] **redistribute rip [level-1 | level-1-2 | level-2] | metric** <*number*> **| metric-type [external | internal] | route-map** <*name*>

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing OSPF routes into IPv4 IS-IS

To redistribute OSPF routes into IPv4 IS-IS, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# redistribute ospf
```

This command configures the PowerConnect to redistribute all OSPF routes into Level-2 IPv4 IS-IS.

Syntax: [no] redistribute ospf [level-1 | level-1-2 | level-2] |
match [external1 | external2 | internal] |
metric <*number*> |
metric-type [external | internal] |
route-map <*name*>

Most of the parameters are the same as the parameters for the **redistribute static** command. However, the **redistribute ospf** command also has the **match external1** | **external2** | **internal** parameter. This parameter specifies the OSPF route type you want to redistribute into IPv4 IS-IS. By default, the **redistribute ospf** command redistributes only internal routes.

- **external1** – An OSPF type 1 external route.
- **external2** – An OSPF type 2 external route.
- **internal** – An internal route calculated by OSPF.

Redistributing BGP4+ routes into IPv4 IS-IS

To redistribute BGP4+ routes into IPv4 IS-IS, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# redistribute bgp
```

This command configures the router to redistribute all its BGP4 routes into Level-2 IPv4 IS-IS.

Syntax: [no] redistribute bgp [level-1 | level-1-2 | level-2] |
metric <*number*> | metric-type [external | internal] |
route-map <*name*>

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing IPv4 IS-IS routes within IPv4 IS-IS

In addition to redistributing routes from other route sources into IPv4 IS-IS, the PowerConnect can redistribute Level 1 IPv4 IS-IS routes into Level 2 IPv4 IS-IS routes, and Level 2 IPv4 IS-IS routes into Level 1 IPv4 IS-IS routes. By default, the device redistributes routes from Level 1 into Level 2.

NOTE
The PowerConnect automatically redistributes Level 1 routes into Level 2 routes, even if you do not enable redistribution.

For example, to redistribute all IPv4 IS-IS routes from Level 2 into Level 1, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv4u)# redistribute isis level-2 into level-1
```

The router automatically redistributes Level-1 routes into Level 2.

Syntax: [no] redistribute isis level-1 into level-2 | level-2 into level-1 [prefix-list <*name*>]

The **level-1 into level-2** | **level-2 into level-1** parameter specifies the direction of the redistribution:

- **level-1 into level-2** – Redistributes Level 1 routes into Level 2. This is the default.
- **level-2 into level-1** – Redistributes Level 2 routes into Level 1.

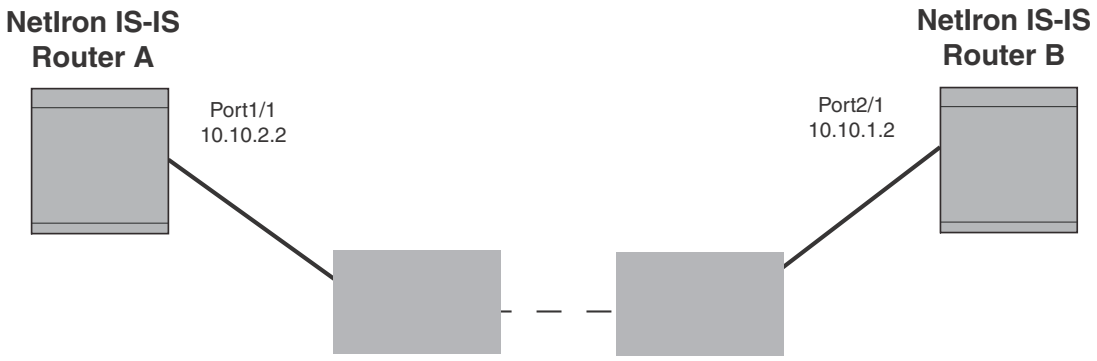
The **prefix-list** <*name*> specifies an IP prefix list.

Configuring IS-IS point-to-point over Ethernet

IS-IS uses its neighbor's MAC address to form an adjacency and stores the neighbors MAC address to recognize the adjacency in the future. This is no problem with directly adjacent routers but can become a problem when adjacency is required between routers that are more than one hop away. To accommodate an IS-IS network with this type of configuration, the IS-IS Point-to-Point over Ethernet feature has been developed.

Using the IS-IS Point-to-Point feature over ethernet, routers that are several hops away or available through an IP GRE tunnel (as described in "Configuring IS-IS over a GRE IP tunnel" on page 960) can form an IS-IS adjacency. It can be used when only two IS's are part of the broadcast network. This feature is configured at the interface level of the routers that are forming an adjacency. For example, Figure 144 shows two PowerConnect routers several hops away from each other that are configured for IS-IS adjacency.

FIGURE 144 IS-IS Point-to-Point configuration



You can use the commands in the following configurations to enable the IS-IS Point-to-Point feature:

PowerConnect IS-IS Router A configuration

To configure PowerConnect IS-IS Router A for the IS-IS Point-to-Point feature use the following commands.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# ip router isis
NetIron(config-if-e10000-1/1)# ip address 10.10.2.2
NetIron(config-if-e10000-1/1)# isis point-to-point
```

PowerConnect IS-IS Router B configuration

To configure PowerConnect IS-IS Router B for the IS-IS Point-to-Point feature use the following commands.

```
NetIron(config)# interface ethernet 2/1
NetIron(config-if-e10000-2/1)# ip router isis
NetIron(config-if-e10000-2/1)# ip address 10.10.1.2
NetIron(config-if-e10000-2/1)# isis point-to-point
```

Syntax: [no] isis point-to-point

NOTE
Point-to-Point is the default setting for POS interfaces and cannot be changed.

Displaying IS-IS point-to-point configuration

Use the **show isis interface** command to determine if IS-IS point-to-point is configured on an interface. In the example below, the lines in bold identify IS-IS point-to-point configuration.

```
NetIron# show isis interface
Total number of IS-IS Interfaces: 2
Interface : v128 Local Circuit Number: 0000000c
   Circuit Type : PTP Circuit Mode : LEVEL-1-2
   Circuit State: UP Passive State: FALSE
   MTU : 1497
   Level-1 Metric: 10, Level-1 Priority: 64
   Circuit State Changes: 1 Circuit Adjacencies State Changes: 1
   Rejected Adjacencies: 0
   Circuit Authentication Fails: 0 Bad LSP 0
   Control Messages Sent: 45600 Control Messages Received: 6778
   IP Enabled: TRUE
   IP Address and Subnet Mask:
    128.1.1.1 255.255.255.0
   IPv6 Enabled: FALSE
```

To determine if IS-IS point-to-point link is being used by ISs, use the **show isis neighbor** command.

```
NetIron# show isis neighbor
System Id   Interface  SNPA           State Holdtime Type Pri StateChgeTime
SFO-RX16    eth1/1     0004.badb.0eee UP    10       ISL2 64  0 :5 :5 :12
SFO-RX16    eth1/1     0004.badb.0eee UP    10       ISL1 64 0 :5 :5 :12
SFO-RX16    ve 128     0900.2b00.0005 UP    30       PTPT 127 0 :4 :46:59
```

Configuring IS-IS over a GRE IP tunnel

As described in "Configuring IS-IS point-to-point over Ethernet" on page 958, IS-IS adjacency can be established over ethernet between routers that are more than one hop away using the IS-IS Point-to-Point feature. IS-IS over a GRE IP tunnel extends this capability by allowing you to configure IS-IS adjacency between routers on either end of a GRE IP tunnel. To configure IS-IS over a GRE IP Tunnel you must configure the following:

- Configure the routers that you want to establish adjacency for IS-IS point-to-point as described in "Configuring IS-IS point-to-point over Ethernet" on page 958.
- Configure a GRE IP Tunnel.
- Configure the routers used for the GRE IP Tunnel for IS-IS using the **router isis** command.
- Configure the tunnel interfaces on the routers used for the GRE IP Tunnel for IS-IS point-to-point using the **isis point-to-point** command.

Configuration considerations

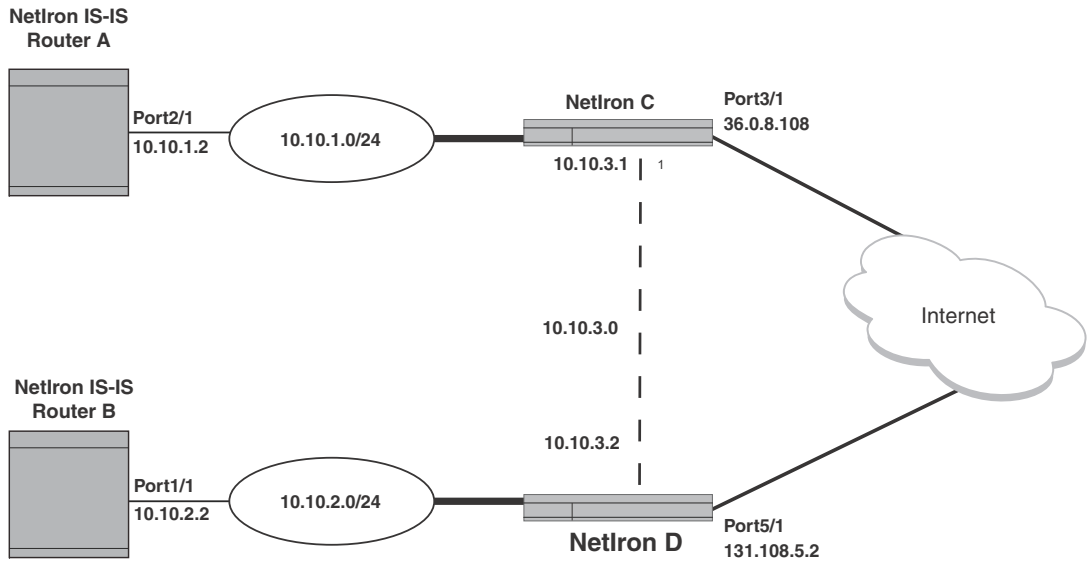
The configuration considerations are as follows:

- When a GRE tunnel is configured, you cannot configure the same routing protocol on the tunnel through which the router learns the route to the tunnel destination. For example, if a router learns the tunnel destination route through the OSPF protocol, you cannot configure the OSPF protocol on the same Tunnel and vice-versa. When a tunnel has OSPF configured, the router cannot learn the tunnel destination route through OSPF. This will cause the system to become unstable.
- When you have keepalive configured on both sides of a GRE tunnel, we recommend that you disable the tunnel before changing any tunnel configurations. You can then re-enable the tunnel to restore it to normal functionality.
- When configuring a GRE IP Tunnel, the router must be configured with one of the following CAM Profiles: ipv4, ipv6, mpls-l3vpn, ipv4-vpn, multi-service-2 or mpls-l3vpn-2

Configuring IS-IS over a GRE IP tunnel

Figure 145 displays a network configured for IS-IS over a GRE IP tunnel. In the example, PowerConnect IS-IS Router A and PowerConnect IS-IS Router B are configured for adjacency. Routers PowerConnect C and PowerConnect D are configured with a GRE IP tunnel. Following the illustration are examples of the configurations required for IS-IS over a GRE IP tunnel.

FIGURE 145 IS-IS over a GRE IP tunnel



The following examples describe the configurations that support IS-IS over a GRE IP tunnel for each of the routers in Figure 145.

PowerConnect IS-IS Router A configuration

To configure PowerConnect IS-IS Router A for the IS-IS Point-to-Point feature use the following commands.

```
NetIron(config)# interface ethernet 2/1
NetIron(config-if-e10000-2/1)# ip router isis
NetIron(config-if-e10000-2/1)# ip address 10.10.1.2
NetIron(config-if-e10000-2/1)# isis point-to-point
```

PowerConnect IS-IS Router B configuration

To configure PowerConnect IS-IS Router B for the IS-IS Point-to-Point feature use the following commands.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# ip router isis
NetIron(config-if-e10000-1/1)# ip address 10.10.2.2
NetIron(config-if-e10000-1/1)# isis point-to-point
```

PowerConnect C configuration

To configure the PowerConnect C router for the IS-IS over a GRE IP tunnel feature, use the following commands.

```
NetIron(config)# router isis
NetIron(config-isis-router)# exit
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1)tunnel source 36.0.8.108
NetIron(config-tnif-1)tunnel destination 131.108.5.2
NetIron(config-tnif-1)tunnel mode gre ip
NetIron(config-tnif-1)isis point-to-point
NetIron(config-tnif-1)ip address 10.10.3.1/24
NetIron(config-tnif-1)exit
NetIron(config) ip route 10.10.2.0/24 10.10.3.1
```

PowerConnect D configuration

To configure the PowerConnect D router for the S-IS over a GRE IP tunnel feature, use the following commands.

```
NetIron(config)# router isis
NetIron(config-isis-router)# exit
NetIron(config)# interface tunnel 1
NetIron(config-tnif-1) tunnel source ethernet 5/1
NetIron(config-tnif-1) tunnel destination 36.0.8.108
NetIron(config-tnif-1) tunnel mode gre ip
NetIron(config-tnif-1) isis point-to-point
NetIron(config-tnif-1) ip address 10.10.3.2/24
NetIron(config-tnif-1) exit
NetIron(config) ip route 10.10.1.0/24 10.10.3.1
```

Displaying IS-IS over GRE IP tunnel

You can use the **show isis interface** command to determine if IS-IS point-to-point is configured on a tunnel interface. In the example below, the lines in bold identify IS-IS point-to-point configuration in the gre_tnl 1 interface.

```
NetIron# show isis interface
Total number of IS-IS Interfaces: 2
Interface : gre_tnl 1
   Circuit State: UP Circuit Mode: LEVEL-1-2
   Circuit Type : PTP Passive State: FALSE
   Circuit Number: 0x02, MTU: 1497
   Level-1 Metric: 10, Level-1 Priority: 64
   Level-1 Auth-mode: None
   Level-2 Auth-mode: None
   Level-1 Metric: 10, Level-1 Priority: 50
```

```
Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
Level-1 Designated IS: MLXe1-02 Level-1 DIS Changes: 0
Level-2 Metric: 10, Level-2 Priority: 50
Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
Level-2 Designated IS: MLX2-02 Level-2 DIS Changes: 0
Circuit State Changes: 1 Circuit Adjacencies State Changes: 1
Rejected Adjacencies: 0
Circuit Authentication L1 failures: 0
Circuit Authentication L2 failures: 0
Bad LSPs 0
Control Messages Sent: 318 Control Messages Received: 229
IP Enabled: TRUE
IP Address and Subnet Mask:
 50.50.50.20         255.255.255.0
IPv6 Enabled: FALSE
```

To determine if IS-IS point-to-point link is being used by ISs, use the **show isis neighbor** command. In the example below, the line in bold identifies a point-to-point configuration on the MLXe1 system for the gre_tnl 1 interface.

```
NetIron# show isis neighbor
Total number of IS-IS Neighbors: 3
System Id      Interface  SNPA           State Holdtime Type  Pri    StateChgeTime
0000.0000.0004 eth 6/2    0000.0576.4805  UP    30       ISL1  0   0  :0 :8 :42
0000.0000.0004 eth 6/2    0000.0576.4805  UP    30       ISL2  0   0  :0 :8 :42
MLXe1           gre_tnl 1  0900.2b00.0005  UP    30       PTPT  127 0  :0 :9 :16
```

You can use the **show ip route isis** command to determine if next hop is a tunnel. For example.

```
NetIron# show ip route isis
Type Codes - B:BGP D: Connected I: ISIS S: Static R: RIP O:O SPF; Cost - Dist/Metric
        Destination        Gateway         Port         Cost       Type
1       30.30.30.0/24      50.50.50.10     gre_tnl 1    115/20     IL1
2       100.100.100.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
3       100.100.101.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
4       100.100.102.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
5       100.100.103.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
6       100.100.104.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
7       100.100.105.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
8       100.100.106.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
9       100.100.107.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
```

IS-IS Non-Stop Routing

Overview

NOTE
IS-IS Non-Stop Routing (NSR) is applicable only to IPv4 routes computed by IS-IS and does not apply to IPv6 routes.

IS-IS Non-Stop Routing (NSR) enables the IS-IS router to maintain topology and data flow to avoid re-convergence in the network during a processor switchover or hitless-reload event. The IS-IS Bidirectional Forwarding Detection (BFD) sessions survive the switchover and hitless-reload conditions. In general, a router restart causes its peer to remove the routes originated from the router and reinstalls them. This IS-IS NSR feature enables the router to maintain neighborship and LSA database with its peer on the event of a router restart.

In IS-IS NSR, the processor switchovers and the hitless-reloads are treated the same as they are during startup and the overload bit is set in the same way as it is after a reboot. For more information on overload bit setup, refer to "Setting the overload-bit" in Chapter 31.

NOTE
IS-IS NSR is independent of Graceful Restart (GR) and GR help role mechanisms.

Limitations

- The IS-IS over GRE tunnel feature does not support IS-IS NSR. The GRE tunnel interface types are not supported.
- The IS-IS shortcuts are not supported because they depend on the MPLS tunnel.
- If the IS-IS hellos are forwarded at Layer 2 and the switch executes a hitless-reload, hellos will not be forwarded for a brief time. The IS-IS adjacencies are lost for 12 seconds and there will be data traffic loss.
- The configuration events that occur close to switchover or hitless-reload may get lost due to CLI synchronization issues.
- The neighbor or interface state changes close to switchover or hitless-reload cannot be handled.
- The IS-IS neighbor hold timer is restarted upon IS-IS NSR switchover or hitless-reload.
- The traffic counters are not synchronized because the neighbor and LSP database counters are recalculated on the standby module during synchronization.
- With IS-IS NSR enabled, after switchover or hitless-reload to standby MP, IS-IS routes, LSP database and neighbor adjacencies are maintained so that there will be no loss of existing traffic to the IS-IS destinations.
- The IS-IS NSR hitless failover event may not be completely invisible to the network because, after switchover, additional flooding of CSNP packets will occur in the directly connected neighbors.

Enabling and disabling IS-IS NSR

To globally enable IS-IS NSR, enter the following commands.

```
PowerConnect(config)# router isis
PowerConnect(config-isis-router)# nonstop-routing
```

To globally disable IS-IS NSR, enter the following commands.

```
PowerConnect(config)# router isis
PowerConnect(config-isis-router)# no nonstop-routing
```

Syntax: [no] nonstop-routing

Displaying the IS-IS NSR status

To display the IS-IS NSR status, enter the following command.

```
PowerConnect(config-isis-router)# show isis
IS-IS Routing Protocol Operation State: Enabled
 IS-Type: Level-2
 System ID: cccc.bbbb.aaaa
 Manual area address(es):
   22.6666
 Level-1-2 Database State: On
 Administrative Distance: 210
 Maximum Paths: 4
 Default redistribution metric: 0
 Protocol Routes redistributed into IS-IS:
    None
 Number of Routes redistributed into IS-IS: 0
 Level-1 Auth-mode: None
 Level-2 Auth-mode: None
 Metric Style Supported for Level-1: Narrow
 Metric Style Supported for Level-2: Narrow
 IS-IS Partial SPF Optimizations: Enabled
 Timers:
    L1 SPF: Max-wait 5s Init-wait 5000ms Second-wait 5000ms
    L2 SPF: Max-wait 5s Init-wait 5000ms Second-wait 5000ms
    L1 SPF will run in 800msec
    L2 SPF is not scheduled
    PSPF: Max-wait 5000ms Init-wait 2000ms Second-wait 5000ms
    PSPF will run in 300msec
    LSP: max-lifetime 45s, refresh-interval 7s, gen-interval 10s
         retransmit-interval 5s, lsp-interval 33ms
    SNP: csnp-interval 10s, psnp-interval 2s
 Global Hello Padding : Enabled
 Global Hello Padding For Point to Point Circuits: Enabled
 Ptpt Three Way HandShake Mechanism: Enabled
 BGP Ipv4 Converged: FALSE, Ipv6 Converged: FALSE
 IS-IS Traffic Engineering Support: Disabled
 No ISIS Shortcuts Configured
 BFD: Disabled
 NSR: Enabled
    NSR State: Normal
    Standby MP: Ready
    Sync State: Enabled
```

```
Interfaces with IPv4 IS-IS configured:
   ethernet 2/1 ve 20 ve 165 loopback 1 loopback 2 loopback 3
```

The following table describes the output of the **show isis** command.

TABLE 153 Output from the **show isis** command

This field...	Displays...
IS-IS Routing Protocol Operation State	This field indicates the operating state of IS-IS and the possible states includes the following: • Enabled – IS-IS is enabled. • Disabled – IS-IS is disabled.
IS-Type	This field indicates the intermediate system type and the possible types includes the following: • Level 1 only – The PowerConnect routes traffic only within the area in which it resides. • Level 2 only – The PowerConnect routes traffic between areas of a routing domain. • Level 1-2 – The PowerConnect routes traffic within the area in which it resides and between areas of a routing domain.
System ID	This field indicates the unique IS-IS router ID. Typically, the router base MAC address is used as the system ID.
Manual area address(es)	This field indicates the Area address(es) of the PowerConnect router.
Level-1-2 Database State	This field indicates the state of the Level 1-2 Database: • On • Off
Administrative Distance	This field specifies the current setting of the IS-IS administrative distance.
Maximum Paths	This field specifies the number of paths IS-IS can calculate and install in the forwarding table.
Default redistribution metric	This field specifies the value of the default redistribution metric, which is the IS-IS cost of redistributing the route into IS-IS.
Number of Routes redistributed into IS-IS	This field specifies the number of routes distributed into IS-IS.
Level-1 Auth-mode	This field indicates one of the following authentication modes set for Level-1 on the router: • None • md5 • cleartext
Level-2 Auth-mode	This field indicates one of the following authentication modes set for Level-2 on the router: • None • md5 • cleartext
Metric Style Supported for Level-1	This field indicates the metric style supported for Level-1 and the following values are supported: • Wide – Wide Metric Style • Narrow – Narrow Metric Style

TABLE 153 Output from the **show isis** command (Continued)

This field...	Displays...
Metric Style Supported for Level-2	This field indicates the metric style supported for Level-2 and the following values are supported: • Wide – Wide Metric Style • Narrow – Narrow Metric Style
IS-IS Partial SPF Optimizations	This field indicates the IS-IS partial SPEF optiomization and the parameter can contain one of the following values: • Enabled • Disabled
Timers: L1 or L2 SPF:	The following values are displayed individually for IS-IS levels 1 and 2.
max-wait	This fields indicates the maximum time gap that will occur between running of SPF calculations. It is the value configured as the spf-max-wait variable in the **spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
Init-wait	This fields indicates the initial time gap between an SPF event and the first running of SPF. This value reflects the spf-initial-time variable that is configured using the **spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
Second-wait	This fields indicates the interval between the first running of SPF and the first recalculation of the SPF tree. If this optional value is configured, it will be doubled with each recalculation of the SPF tree until the value is equal to the max-wait value This value reflects the spf-second-wait variable that is configured using the **spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
SPF run status.	This field is not specifically labeled but it is displayed directly under the SPF timers. It can be any of the three values shown below: • SPF is running • SPF will run in <*sec*> where the <*sec*> variable is a value in seconds until the next time that SPF will be run. • SPF is not scheduled
Timers: PSPF:	
max-wait	This fields indicates the maximum time gap that will occur between running of PSPF calculations. It is the value configured as the max-wait value in the **partial-spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
Init-wait	This fields indicates the initial time gap between the wait time after an LSP change until the first PSPF calculation. This value reflects the initial-wait variable that is configured using the **partial-spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
Second-wait	This fields indicates the wait time between the first and second PSPF calculations. If this optional value is configured, it will be doubled with each PSPF recalculation until the value is equal to the max-wait value This value reflects the second-wait variable that is configured using the **partial-spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.

TABLE 153 Output from the **show isis** command (Continued)

This field...	Displays...
PSPF run status.	This field is not specifically labeled but it is displayed directly under the PSPF timers. It can be any of the three values shown below: • PSPF is running • PSPF will run in <sec> where the <sec> variable is a value in seconds until the next time that PSPF will be run. • PSPF is not scheduled
Timers: LSP:	
max-lifetime	This fields indicates the maximum number of seconds an unrefreshed LSP can remain in the PowerConnect router's LSP database. The default value is 1000 sec.
refresh-interval	This fields indicates the maximum number of seconds that a PowerConnect router waits between sending updated LSPs to its IS-IS neighbors. The default value is 1 sec.
gen-interval	This fields indicates the minimum number of seconds that a PowerConnect router waits between sending updated LSPs to its IS-IS neighbors. The default value is 10 sec.
retransmit-interval	This fields indicates the amount of time the PowerConnect router waits before it retransmits LSPs. The default value is 5 sec.
lsp-interval	This fields indicates the rate of transmission (in milliseconds) of the LSPs. The default rate is 33 ms.
Timers: SNP:	
csnp-interval	This fields indicates how often the designated IS sends a CSNP to the broadcast interface. The default value is 10 sec.
psnp-interval	This fields indicates how often the IS sends a PSNP. The default value is 2 sec.
Global Hello Padding	The value can be: • Enabled • Disabled
Global Hello Padding For Point to Point Circuits	The value can be: • Enabled • Disabled
Ptpt Three Way HandShake Mechanism	The value can be: • Enabled • Disabled
IS-IS Traffic Engineering Support	The value can be: • Enabled • Disabled
BFD	The value can be: • Enabled • Disabled

TABLE 153 Output from the **show isis** command (Continued)

This field...	Displays...
Interfaces with IPv4 IS-IS configured	This field specifies the interfaces on which IPv4 IS-IS is configured.
NSR state	This field indicates the state of the IS-IS NSR and takes the following values: • Normal - This indicates that the switchover is either compete or the switchover event is not triggerd. • SwitchOver Detected - This indicates that the switchover event is recognized by the IS-IS. • All Card Done - This is an internal event after which the IS-IS starts sending hellos to its neighbors and schedules SPF. • SPF Run Complete - This indicates that the SPF run and updating of the IS-IS routes to RTM is complete. • Wait for BGP - This event indicates that the IS-IS is waiting for redistribution to complete. After redistribution to IS-IS is complete, the IS-IS NSR state will change to Normal.
Standby MP	This field indicates the standby MP is active, ready, or inactive: • Active - This indicates the standby MP is active. • Inactive - This indicates the standby MP is either down or not present. • Ready - This indicates the standby MP is ready to accept configuration updates or database updates.
Sync State	This field indicates whether the synchronization state is enabled or disabled. The state changes depending on whether or not the **Non Stop-Routing** command is configured under the router IS-IS.
Interfaces with IPv4 IS-IS configured	This field specifies the interfaces configured with IPv4 IS-IS.

Configuring ISIS properties on an interface

This section describe the IS-IS parameters for an interface.

Disabling and enabling IS-IS on an interface

In addition to enabling IS-IS globally, you also must enable the protocol on the individual interfaces connected to ISs or ESs. To enable IS-IS locally on specific interfaces, enter commands such as the following.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-1/1)# ip router isis
NetIron(config-if-1/1)# exit
NetIron(config)# interface pos 2/3
NetIron(config-if-2/3)# ip router isis
```

These commands enable IS-IS on ports 1/1 and 2/3. The NET configured above (at the IS-IS configuration level) applies to both interfaces.

Syntax: [no] ip router isis

Disabling or re-enabling formation of adjacencies

When you enable IS-IS on any type of interface except a loopback interface, the interface also is enabled to send advertisements and form an adjacency with an IS at the other end of the link by default. Adjacency formation and advertisements are disabled by default on loopback interfaces.

You can enable or disable adjacency formation and advertisements on an interface.

NOTE
The PowerConnect advertises an IS-IS interface to its area regardless of whether adjacency formation is enabled.

To disable IS-IS adjacency formation on an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# isis passive
```

This command disables IS-IS adjacency formation on port 2/8. The device still advertises this IS-IS interface into the area, but does not allow the port to form an adjacency with the IS at the other end of the link.

Syntax: [no] isis passive

Setting the priority for designated IS election

The priority of an IS-IS interface determines the priority of the interface for being elected as a Designated IS. Level-1 has a Designated IS and Level-2 has a Designated IS. The Level-1 and Level-2 Designated ISs are independent, although the same device can become both the Level-1 Designated IS and the Level-2 Designated IS.

By default, the Level-1 and Level-2 priority is 64. You can configure an interface's priority to a value from 0 – 127. You can configure the same priority for both Level-1 and Level-2 or you can configure a different priority for each level. In case of a tie (if two or more devices have the highest priority within a given level), the device with the highest MAC address becomes the Designated IS for that level.

NOTE
You can set the IS-IS priority on an individual interface basis only. You cannot set the priority globally.

To set the IS-IS priority on an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# isis priority 127
```

This command sets the IS-IS priority on port 1/1 to 127. Since the command does not specify Level-1 or Level-2, the new priority setting applies to both IS-IS levels.

Syntax: [no] isis priority <*num*> **[level-1 | level-2]**

The <*num*> parameter specifies the priority and can be from 0 – 127. A higher numeric value means a higher priority. The default is 64.

The **level-1 | level-2** parameter applies the priority to Level-1 only or Level-2 only. By default, the priority is applied to both levels.

Limiting access to adjacencies with a neighbor

In addition to limiting access to an area (level-1) or domain (level-2), you can limit access to forming an IS-IS adjacency on a specific interface by entering a password at the interface configuration level. To enter this password, enter a command such as the following.

```
NetIron(config)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# isis password my-password
```

Syntax: [no] isis password <*string*>

The <*string*> parameter specifies the password. You can enter an alphanumeric string up to 80 characters long. The password can contain blank spaces. If you use a blank space in the password, you must use quotation marks (“ “) around the entire password; for example, **isis password “admin 2”.**

Changing the IS-IS level on an interface

The section “Changing the IS-IS level globally” on page 945 explains how to change the IS-IS level globally. By default, a PowerConnect can operate as both a Level-1 and IS-IS Level-2 router. You can change the IS-IS type on an individual interface to be Level-1 only or Level-2 only. You also can reset the type to both Level-1 and Level-2.

NOTE
If you change the IS-IS type on an individual interface, the type you specify must also be specified globally. For example, if you globally set the type to Level-2 only, you cannot set the type on an individual interface to Level-1. The software accepts the setting but the setting does not take effect.

To change the IS-IS type on a specific interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# isis circuit-type level-1
```

Syntax: [no] isis circuit-type level-1 | level-1-2 | level-2

The **level-1 | level-1-2 | level-2** parameter specifies the IS-IS type. If you want to re-enable support for both IS-IS types, re-enter the command you entered to change the IS-IS type, and use “no” in front of the command.

Disabling and enabling hello padding on an interface

The section “Globally disabling or re-enabling hello padding” on page 949 explains what hello padding is, why it is important and how to globally disable or enable it on a device. You can also disable hello padding on a specific interface by entering commands such as the following.

```
NetIron(config)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# no isis hello padding
```

Syntax: [no] isis hello padding

By default, hello padding is enabled. Enter the **no** form of the command to disable hello padding.

Changing the hello interval

The hello interval controls how often an IS-IS interface sends hello messages to its IS-IS neighbors. The default interval is 10 seconds for Level-1 and Level-2. You can change the hello interval for one or both levels to a value from 1 – 65535 seconds.

To change the hello interval for Ethernet interface 2/8, enter commands such as the following.

```
NetIron(config)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# isis hello-interval 20
```

This command changes the hello interval to 20 seconds. By default, the change applies to Level-1 and Level-2.

Syntax: [no] isis hello-interval <*num*> **[level-1 | level-2]**

The <*num*> parameter specifies the interval, and can be from 1 – 65535 seconds. The default is 10 seconds.

The **level-1 | level-2** parameter applies the change to only the level you specify. If you do not use this parameter, the change applies to both levels.

Changing the hello multiplier

The hello multiplier is the number by which an IS-IS interface multiplies the hello interval to obtain the hold time for Level-1 and Level-2 IS-to-IS hello PDUs. The default multiplier is 3. You can set the multiplier to a value in the range 3 – 1000.

To change the hello multiplier for Ethernet interface 2/8, enter commands such as the following.

```
NetIron(config)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# isis hello-multiplier 50
```

This command changes the hello interval to 50. By default, the change applies to both Level-1 and Level-2.

Syntax: [no] isis hello-multiplier <*num*> **[level-1 | level-2]**

The <*num*> parameter specifies the multiplier, and can be from 3 – 1000. The default is 3.

The **level-1 | level-2** parameter applies the change to only the level you specify. If you do not use this parameter, the change applies to both levels.

Changing the metric added to advertised routes

When the PowerConnect originates an IS-IS route or calculates a route, the PowerConnect adds a metric (cost) to the route. Each IS-IS interface has a separate metric value. The default is 10.

The PowerConnect applies the interface-level metric to routes originated on the interface and also when calculating routes. The PowerConnect does not apply the metric to link-state information that the PowerConnect receives from one IS and floods to other ISs.

The default interface metric is 10. You can change the metric on an individual interface to a value in one of the following ranges:

- 1 – 63 for the narrow metric style (the default metric style for IPv4 ISIS)
- 1 – 16777215 for the wide metric style (the default metric style for IPv4 ISIS)

NOTE
If the metric value you want to use is higher than 63 but you have not changed the metric style to wide, change the metric style first, then set the metric. The IS-IS neighbors that will receive the advertisements also must be enabled to receive wide metrics.

To change the IS-IS metric on an interface, use the following CLI method.

```
NetIron(config)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# isis metric 15
```

Syntax: [no] isis metric <*num*> **[level-1 | level-2]**

The <*num*> parameter specifies the metric. The range of values you can specify depends on the metric style. You can specify 1 – 63 for the narrow metric style or 1 – 16777215 for the wide metric style. The default in either case is 10.

The **level-1 | level-2** parameter applies the change to only the level you specify. If you do not use this parameter, the change applies to both levels.

Displaying IPv4 IS-IS information

You can display the following information:

- General IS-IS Information – "Displaying ISIS general information" on page 972
- The active configuration (the IS-IS commands in the running-config) – refer to "Displaying the IS-IS configuration in the running-config" on page 976
- Name mappings – "Displaying the name mappings" on page 976
- Neighbor information – "Displaying neighbor information" on page 977
- Neighbor adjacency changes – "Displaying IS-IS Syslog messages" on page 979
- Interface information – "Displaying interface information" on page 979
- Route information – "Displaying route information" on page 983
- LSP database entries – "Displaying LSP database entries" on page 984
- Traffic statistics – "Displaying traffic statistics" on page 988
- Error statistics – "Displaying error statistics" on page 989
- IS-IS Log – "Displaying the IS-IS SPF Log" on page 992

Displaying ISIS general information

To display general IPv4 IS-IS information, enter the following command at any CLI level.

```
NetIron#show isis
 IS-IS Routing Protocol Operation State: Enabled
IS-Type: Level-1-2
System ID: 1111.1111.1111
Manual area address(es):
47
Level-1-2 Database State: On
Administrative Distance: 115
Maximum Paths: 4
Default redistribution metric: 0
Protocol Routes redistributed into IS-IS:
```

```
Static
Number of Routes redistributed into IS-IS: 11
Level-1 Auth-mode: None
Level-2 Auth-mode: None
Metric Style Supported for Level-1: Wide
Metric Style Supported for Level-2: Wide
IS-IS Partial SPF Optimizations: Enabled
Timers:
L1 SPF: Max-wait 120s Init-wait 100ms Second-wait 120000ms
L2 SPF: Max-wait 100s Init-wait 100ms Second-wait 100000ms
L1 SPF is not scheduled
L2 SPF is not scheduled
PSPF: Max-wait 120000ms Init-wait 120000ms Second-wait 120000ms
PSPF is not scheduled
 LSP: max-lifetime 1200s, refresh-interval 900s, gen-interval 10s
retransmit-interval 5s, lsp-interval 33ms
SNP: csnp-interval 10s, psnp-interval 2s
Global Hello Padding : Enabled
Global Hello Padding For Point to Point Circuits: Enabled
Ptpt Three Way HandShake Mechanism: Enabled
IS-IS Traffic Engineering Support: Disabled
BFD: Disabled
Interfaces with IPv4 IS-IS configured:
eth 1/1
```

Syntax: show isis

This display shows the following information.

TABLE 154 IS-IS neighbor information

This field...	Displays...
IS-IS Routing Protocol Operation State	The operating state of IS-IS. Possible states include the following: • Enabled – IS-IS is enabled. • Disabled – IS-IS is disabled.
IS-Type	The intermediate system type. Possible types include the following: • Level 1 only – The PowerConnect routes traffic only within the area in which it resides. • Level 2 only – The PowerConnect routes traffic between areas of a routing domain. • Level 1-2 – The PowerConnect routes traffic within the area in which it resides and between areas of a routing domain.
System ID	The unique IS-IS router ID. Typically, the router's device's base MAC address is used as the system ID.
Manual area address(es)	Area address(es) of the PowerConnect router.
Level-1-2 Database State	The state of the Level 1-2 Database: • On • Off
Administrative Distance	The current setting of the IS-IS administrative distance.
Maximum Paths	The number of paths IS-IS can calculate and install in the forwarding table
Default redistribution metric	The value of the default redistribution metric, which is the IS-IS cost of redistributing the route into IS-IS.

TABLE 154 IS-IS neighbor information (Continued)

This field...	Displays...
Number of Routes redistributed into IS-IS	The number of routes distributed into IS-IS.
Level-1 Auth-mode	One of the following authentication modes set for Level-1 on the router: • None • md5 • cleartext
Level-2 Auth-mode	One of the following authentication modes set for Level-2 on the router: • None • md5 • cleartext
Metric Style Supported for Level-1	The following values are supported: • Wide – Wide Metric Style • Narrow – Narrow Metric Style
Metric Style Supported for Level-2	The following values are supported: • Wide – Wide Metric Style • Narrow – Narrow Metric Style
IS-IS Partial SPF Optimizations	This parameter can contain one of the following values: • Enabled • Disabled
Timers: L1 or L2 SPF:	These values are displayed individually for IS-IS levels 1 and 2.
max-wait	The maximum time gap that will occur between running of SPF calculations. It is the value configured as the spf-max-wait variable in the **spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
Init-wait	The initial time gap between an SPF event and the first running of SPF. This value reflects the spf-initial-time variable that is configured using the **spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
Second-wait	The interval between the first running of SPF and the first recalculation of the SPF tree. If this optional value is configured, it will be doubled with each recalculation of the SPF tree until the value is equal to the max-wait value This value reflects the spf-second-wait variable that is configured using the **spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
SPF run status.	This field is not specifically labeled but is displayed directly under the SPF timers.) It can any of the three values shown below: • SPF is running • SPF will run in <*sec*> where the <*sec*> variable is a value in seconds until the next time that SPF will be run. • SPF is not scheduled
Timers: PSPF:	
max-wait	The maximum time gap that will occur between running of PSPF calculations. It is the value configured as the max-wait value in the **partial-spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.

TABLE 154 IS-IS neighbor information (Continued)

This field...	Displays...
Init-wait	The initial time gap between the wait time after an LSP change until the first PSPF calculation. This value reflects the initial-wait variable that is configured using the **partial-spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
Second-wait	The wait time between the first and second PSPF calculations. If this optional value is configured, it will be doubled with each PSPF recalculation until the value is equal to the max-wait value This value reflects the second-wait variable that is configured using the **partial-spf-interval** command as described in "Configuring the IS-IS PSPF exponential back-off feature" on page 948.
PSPF run status.	This field is not specifically labeled but is displayed directly under the PSPF timers. It can any of the three values shown below: • PSPF is running • PSPF will run in <*sec*> where the <*sec*> variable is a value in seconds until the next time that PSPF will be run. • PSPF is not scheduled
Timers: LSP:	
max-lifetime	The maximum number of seconds an unrefreshed LSP can remain in the PowerConnect router's LSP database. The default value is 1000 sec.
refresh-interval	The maximum number of seconds that a PowerConnect router waits between sending updated LSPs to its IS-IS neighbors. The default value is 1 sec.
gen-interval	The minimum number of seconds that a PowerConnect router waits between sending updated LSPs to its IS-IS neighbors. The default value is 10 sec.
retransmit-interval	The amount of time the PowerConnect router waits before it retransmits LSPs. The default value is 5 sec.
lsp-interval	The rate of transmission (in milliseconds) of the LSPs. The default rate is 33 ms.
Timers: SNP:	
csnp-interval	How often the designated IS sends a CSNP to the broadcast interface. The default value is 10 sec.
psnp-interval	How often the IS sends a PSNP. The default value is 2 sec.
Global Hello Padding	This value can be: • Enabled • Disabled
Global Hello Padding For Point to Point Circuits	This value can be: • Enabled • Disabled
Ptpt Three Way HandShake Mechanism	This value can be: • Enabled • Disabled

TABLE 154 IS-IS neighbor information (Continued)

This field...	Displays...
IS-IS Traffic Engineering Support	This value can be: • Enabled • Disabled
BFD	This value can be: • Enabled • Disabled
Interfaces with IPv4 IS-IS configured	Interfaces on which IPv4 IS-IS is configured.

Displaying the IS-IS configuration in the running-config

You can display the global IS-IS configuration commands that are in effect on the PowerConnect using the following CLI method.

NOTE
The running-config does not list the default values. Only commands that change a setting or add configuration information are displayed.

To list the global IS-IS configuration commands in the PowerConnect's running-config, enter the following command at any level of the CLI.

```
NetIron# show isis config

router isis
 net  20.00e0.5200.0001.00
end
```

The running-config shown in this example contains the command that enables IS-IS and a command that configures a NET.

To display the interface configuration information in the running-config, enter one of the following commands at any level of the CLI:

- **show running-config**
- **write terminal**

Syntax: show isis config

Displaying the name mappings

To display the mappings, enter the following command at any level of the CLI.

```
NetIron# show isis hostname
Total number of entries in IS-IS Hostname Table: 1
   System ID        Hostname            * = local IS
 * bbbb.cccc.dddd   MLXe
```

Syntax: show isis hostname

The table in this example contains one mapping, for this PowerConnect. The PowerConnect's IS-IS system ID is "bbbb.cccc.dddd" and its hostname is "MLXe". The display contains one entry for each IS that supports name mapping.

NOTE
Name mapping is enabled by default. When name mapping is enabled, the output of the **show isis database**, **show isis interface**, and **show isis neighbor** commands uses the host name instead of the system ID. To disable mapping so that these displays use the system ID instead, refer to "Disabling or re-enabling display of hostname" on page 945.

Displaying neighbor information

To display IS-IS neighbor information, enter the following command at any level of the CLI.

```
NetIron# show isis neighbor
Total number of IS-IS Neighbors: 2
System ID      Interface   SNPA           State Holdtime Type Pri StateChgeTime
00e0.52b5.7800 Ether2/4    00e0.52b5.7843 UP    10       ISL2 64  0  :0 :16:8
00e0.52b5.7800 Ether2/4    00e0.52b5.7843 UP    10       ISL1 64  0  :0 :16:8
```

Syntax: show isis neighbor [detail]

The **detail** option displays more details for each neighbor.

This display shows the following information.

TABLE 155 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	The number of ISs with which the PowerConnect has formed IS-IS adjacencies.
System ID	The System ID of the neighbor or the hostname of the neighbor.
Interface	The PowerConnect port or virtual interface attached to the neighbor.
SNPA	The Subnetwork Point of Attachment (SNPA), which is the MAC address of the PowerConnect port or virtual interface attached to the neighbor.
State	The state of the adjacency with the neighbor. The state can be one of the following: • DOWN – The adjacency is down. • INIT – The adjacency is being established and is not up yet. • UP – The adjacency is up.
Holdtime	The neighbor's advertised hold time.
Type	The IS-IS type of the adjacency. The type can be one of the following: • ISL1 – Level-1 IS • ISL2 – Level-2 IS • ES – ES **NOTE:** The PowerConnect forms a separate adjacency for each IS-IS type. Thus, if the PowerConnect has both types of IS-IS adjacencies with the neighbor, the display contains a separate row of information for each adjacency.
Pri	The priority of this IS to be elected as the Designated IS in this broadcast network.
StateChgeTime	The amount of time that has passed since the adjacency last changed state.
3-Way Handshake TLV received	
Area Address (es)	

TABLE 155 IS-IS neighbor information (Continued)

This field...	Displays...
Protocols Supported	
IP Address	
IPv6 Address	

To display IS-IS neighbor detail information, enter the following command at any level of the CLI.

```
NetIron(config-if-e10000-1/1)#show isis neighbor detail
Total number of IS-IS Neighbors: 1
System ID      Interface   SNPA           State Holdtime Type Pri StateChgeTime
NetIron           eth 1/1     0900.2b00.0005 UP    15       PTPT 127 0  :0 :0 :5
3-Way HandShake TLV received: circuit-id 1
Area Address(es): 47
 Protocols Supported: IP IPv6
 IP Address: 10.1.1.1, circuit-id 1
 IPv6 Address: fe80::200:ff:fe01:c000, circuit-id 1
```

The following table describes the contents of the example display.

TABLE 156 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	The number of ISs with which the PowerConnect has formed IS-IS adjacencies.
System ID	The System ID of the neighbor or the hostname of the neighbor.
Interface	The PowerConnect port or virtual interface attached to the neighbor.
SNPA	The Subnetwork Point of Attachment (SNPA), which is the MAC address of the PowerConnect port or virtual interface attached to the neighbor.
State	The state of the adjacency with the neighbor. The state can be one of the following: • DOWN – The adjacency is down. • INIT – The adjacency is being established and is not up yet. • UP – The adjacency is up.
Holdtime	The neighbor's advertised hold time.
Type	The IS-IS type of the adjacency. The type can be one of the following: • ISL1 – Level-1 IS • ISL2 – Level-2 IS • ES – ES **NOTE:** The PowerConnect forms a separate adjacency for each IS-IS type. Thus, if the PowerConnect has both types of IS-IS adjacencies with the neighbor, the display contains a separate row of information for each adjacency.
Pri	The priority of this IS to be elected as the Designated IS in this broadcast network.
StateChgeTime	The amount of time that has passed since the adjacency last changed state.

Displaying IS-IS Syslog messages

When logging is enabled, the PowerConnect generates Syslog messages and SNMP traps for the following IS-IS events:

- Overload state (the PowerConnect entering or leaving the overload state)
- Memory overrun (IS-IS is demanding more memory than is available)

You also can enable the PowerConnect to generate Syslog messages and SNMP traps when an adjacency with a neighbor comes up or goes down. To enable logging of adjacency changes, refer to "Logging adjacency changes" on page 950.

To display Syslog entries, enter the following command at any level of the CLI.

```
NetIron# show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:

Dynamic Log Buffer (50 lines):
00d00h00m42s:N:BGP Peer 192.147.202.10 UP (ESTABLISHED)
00d00h00m18s:N:ISIS L2 ADJACENCY UP 1234.1234.1234 on interface 2/8
00d00h00m08s:N:ISIS L1 ADJACENCY UP 1234.1234.1234 on interface 2/8
00d00h00m08s:N:ISIS L2 ADJACENCY UP 0000.86de.5520 on interface 5/1
00d00h00m00s:I:Warm start
```

The messages in this example indicate that the software has been reloaded (Warm start) and adjacencies between the PowerConnect and three ISs have come up.

Syntax: show logging

Displaying interface information

To display information about the PowerConnect's IS-IS interfaces, enter the **show isis** commands at any level of the CLI, as the examples in this section illustrate.

The following is an example of the **show isis interface** command for an Ethernet Interface module configured for a Circuit Type BCAST.

```
NetIron(config-if-e10000-1/1)#show isis interface
Total number of IS-IS Interfaces: 1
Interface: eth 1/1
Circuit State: UP Circuit Mode: LEVEL-1-2
Circuit Type: BCAST Passive State: FALSE
Circuit Number: 0x01, MTU: 1500
Level-1 Auth-mode: None
Level-2 Auth-mode: None
Level-1 Metric: 10, Level-1 Priority: 64
Level-1 Hello Interval: 5 Level-1 Hello Multiplier: 3
Level-1 Designated IS: mu2-01 Level-1 DIS Changes: 3
Level-2 Metric: 10, Level-2 Priority: 64
Level-2 Hello Interval: 5 Level-2 Hello Multiplier: 3
Level-2 Designated IS: mu2-01 Level-2 DIS Changes: 3
Next IS-IS LAN Level-1 Hello in 1 seconds
Next IS-IS LAN Level-2 Hello in 4 seconds
```

```
Number of active Level-1 adjacencies: 0
Number of active Level-2 adjacencies: 0
Circuit State Changes: 1 Circuit Adjacencies State Changes: 0
Rejected Adjacencies: 0
Circuit Authentication L1 failures: 0
Circuit Authentication L2 failures: 0
Bad LSPs: 0
Control Messages Sent: 63 Control Messages Received: 27
Hello Padding: Enabled
IP Enabled: TRUE
IP Addresses:
10.1.1.2/24
IPv6 Enabled: TRUE
IPv6 Addresses:
1000::1/32
IPv6 Link-Local Addresses:
fe80::200:ff:fe02:c000
MPLS TE Enabled: FALSE
```

The following is an example of the **show isis interface** command for a POS Interface module configured with a Circuit Type: PTP.

```
NetIron#show isis interface
Total number of IS-IS Interfaces: 1
Interface: eth 1/1
Circuit State: UP Circuit Mode: LEVEL-1-2
Circuit Type: PTP Passive State: FALSE
Circuit Number: 0x01, MTU: 1500
Level-1 Auth-mode: None
Level-1 Metric: 10
Level-1 Hello Interval: 5 Level-1 Hello Multiplier: 3
Level-2 Metric: 10
Circuit State Changes: 1 Circuit Adjacencies State Changes: 0
Rejected Adjacencies: 0
Circuit Authentication L1 failures: 0
Bad LSPs: 0
Control Messages Sent: 9 Control Messages Received: 1
Hello Padding: Enabled
IP Enabled: TRUE
IP Addresses:
10.1.1.2/24
IPv6 Enabled: TRUE
IPv6 Addresses:
1000::1/32
IPv6 Link-Local Addresses:
fe80::200:ff:fe02:c000
MPLS TE Enabled: FALSE
```

Syntax: show isis interface [brief | ethernet <*slot-number*>/<*port-number*> **| pos** <*slot-number*>/<*port-number*> **| loopback** <*number*> **| ve** <*number*>**]**

This display shows the following information.

TABLE 157 IS-IS interface information

This field...	Displays...
Total number of IS-IS interfaces	The number of interfaces on which IS-IS is enabled.
Interface	The port or virtual interface number to which the information listed below applies.

TABLE 157 IS-IS interface information (Continued)

This field...	Displays...
Circuit State	The state of the circuit, which can be one of the following: • DOWN • UP
Circuit Mode	The IS-IS type in use on the circuit. The mode can be one of the following: • LEVEL-1 • LEVEL-2 • LEVEL-1-2
Circuit Type	The type of IS-IS circuit running on the interface. The circuit type can be one of the following: • BCAST (broadcast). • PTP (Point-to-Point)
Passive State	The passive state determines whether the interface is allowed to form an IS-IS adjacency with the IS at the other end of the circuit. The state can be one of the following: • FALSE – The passive option is disabled. The interface can form an adjacency with the IS at the other end of the link. • TRUE – The passive option is enabled. The interface cannot form an adjacency, but can still advertise itself into the area.
Circuit Number	The ID that the instance of IS-IS running on the interface applied to the circuit between this interface and the interface at the other end of the link.
MTU	The maximum length supported for IS-IS PDUs sent on this interface.
Level-1 Auth-mode	One of the following authentication modes set for Level-1 on the router: • None • md5 • cleartext
Level-2 Auth-mode	One of the following authentication modes set for Level-2 on the router: • None • md5 • cleartext This parameter is not displayed for interfaces that are configured with a Point-to-Point circuit type. This is because separate Level-2 Hello messages are not sent on Point-to-Point interfaces. Consequently, "Hello Interval", "Hello Multiplier", and "Authentication" parameters should always be configured in the Level-1 mode for Point-to-Point interfaces.
Level-1 Metric	The default-metric value that the PowerConnect inserts in IS-IS Level-1 PDUs for this interface.
Level-1 Priority	The priority of this IS to be elected as the Designated IS for Level-1 in this broadcast network.
Level-1 Hello Interval	The number of seconds the software waits between sending Level-1 hello PDUs to the IS at the other end of the circuit.
Level-1 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time set in Level-1 Hello PDUs sent on the circuit.
Level-1 Designated IS	The NET of the Level-1 Designated IS.
Level-1 DIS Changes	The number of times the NET of the Level-1 Designated IS has changed.

TABLE 157 IS-IS interface information (Continued)

This field...	Displays...
Level-2 Metric	The default-metric value that the PowerConnect inserts in IS-IS Level-2 PDUs for this interface.
Level-2 Priority	The priority of this IS to be elected as the Designated IS for Level-2 in this broadcast network.
Level-2 Hello Interval	The number of seconds the software waits between sending Level-2 Hello messages to the IS at the other end of the circuit.
Level-2 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time set for Level-2 Hello PDUs sent on this circuit. This parameter is not displayed for interfaces that are configured with a Point-to-Point circuit type. This is because separate Level-2 Hello messages are not sent on Point-to-Point interfaces. Consequently, "Hello Interval", "Hello Multiplier", and "Authentication" parameters should always be configured in the Level-1 mode for Point-to-Point interfaces.
Level-2 Designated IS	The NET of the Level-2 Designated IS.
Level-2 DIS Changes	The number of times the NET of the Level-2 Designated IS has changed.
Next IS-IS LAN Level-1 Hello	Number of seconds before next Level-1 Hello PDU will be transmitted by the PowerConnect router.
Next IS-IS LAN Level-2 Hello	Number of seconds before next Level-2 Hello PDU will be transmitted by the PowerConnect router.
Number of active Level-1 adjacencies	The number of ISs with which this interface has an active Level-1 adjacency.
Number of active Level-2 adjacencies	The number of ISs with which this interface has an active Level-2 adjacency.
Circuit State Changes	The number of times the state of the circuit has changed.
Circuit State Adjacencies Changes	The number of times an adjacency has started or ended on this circuit.
Rejected Adjacencies	The number of adjacency attempts by other ISs rejected by the PowerConnect router.
Circuit Authentication L1 failures	The number of times the PowerConnect router rejected a circuit because the authentication did not match the authentication configured for Level-1on the PowerConnect router.
Circuit Authentication L2 failures	The number of times the PowerConnect router rejected a circuit because the authentication did not match the authentication configured for Level-2 on the PowerConnect router. This parameter is not displayed for interfaces that are configured with a Point-to-Point circuit type. This is because separate Level-2 Hello messages are not sent on Point-to-Point interfaces. Consequently, "Hello Interval", "Hello Multiplier", and "Authentication" parameters should always be configured in the Level-1 mode for Point-to-Point interfaces.
Bad LSP	The number of times the interface received a bad LSP from an IS at the other end of the circuit. The following conditions can cause an LSP to be bad: • Invalid checksum • Invalid length • Invalid lifetime value
Control Messages Sent	The number of IS-IS control PDUs sent on this interface.
Control Messages Received	The number of IS-IS control PDUs received on this interface.

TABLE 157 IS-IS interface information (Continued)

This field...	Displays...
Hello Padding:	The Hello Padding configuration, which can be: • Enabled • Disabled
IP Enabled	If set to TRUE, the IP protocol is enabled for this circuit.
IP Address and Subnet Mask	The IP address and subnet mask for this interface.
IPv6 Enabled	If set to TRUE, the IPv6 protocol is enabled for this circuit.
IPv6 Address and Subnet Mask	The IPv6 address and subnet mask for this interface.
Ipv6 Link-Local Addresses	The IPv6 link local address for this interface.
MPLS TE Enabled:	If set to TRUE, MPLS Traffic Engineering protocol is enabled for this circuit.
BFD Enabled:	If set to TRUE, BiDirectional Forwarding Detection is enabled for this circuit.

Displaying route information

To display the routes in the PowerConnect's IS-IS route table, use either of the following methods.

To display information about the routes in the PowerConnect's IS-IS route table, enter the following command at any level of the CLI.

```
NetIron# show isis routes
Total number of IS-IS routes: 173
Destination      Mask             Cost  Type Tag      Flags
1.0.0.0          255.255.255.0    21    L2   00000000 00000242
   Path: 1       Next Hop IP: 4.1.1.1          Interface: 7/1
1.0.0.0          255.255.255.255  30    L2   00000000 00000242
   Path: 1       Next Hop IP: 4.1.1.1          Interface: 7/1
1.0.0.1          255.255.255.255  30    L2   00000000 00000242
   Path: 1       Next Hop IP: 4.1.1.1          Interface: 7/1
1.0.10.0         255.255.255.0    30    L2   00000000 00000242
   Path: 1       Next Hop IP: 4.1.1.1          Interface: 7/1
```

Syntax: show isis routes [ip-address <*subnet-mask*> **| ip-address/prefix]**

You may enter **ip-address** <*subnet-mask*> or **ip-address/prefix** if you want information for a specific route.

Example

```
NetIron# show isis routes 1.0.111.0 255.255.255.0
1.0.111.0        255.255.255.0    21    L2   00000000 00000242
   Path: 1       Next Hop IP: 4.1.1.1          Interface: 7/1
```

This display shows the following information.

TABLE 158 IS-IS route information

This field...	Displays...
Total number of IS-IS routes	The total number of routes in the PowerConnect's IS-IS route table. The total includes Level-1 and Level-2 routes.
Destination	The IP destination of the route.
Mask	The subnet mask for the destination address.

TABLE 158 IS-IS route information (Continued)

This field...	Displays...
Cost	The IS-IS default metric for the route, which is the cost of using this route to reach the next-hop router to this destination.
Type	The route type, which can be one of the following: • L1 – Level-1 route • L2 – Level-2 route
Tag	The tag value associated with the route.
Path	The path number in the table. The IS-IS route table can contain multiple equal-cost paths to the same destination, in which case the paths are numbered consecutively. When IP load sharing is enabled, the PowerConnect can load balance traffic to the destination across the multiple paths.
Next Hop IP	The IP address of the next-hop interface to the destination.
Interface	The PowerConnect interface (port or virtual interface) attached to the next hop.
Flags	Values used by Dell technical support for troubleshooting.

Displaying LSP database entries

Use the following methods to display summary or detailed information about the entries in the LSP database.

NOTE
The PowerConnect maintains separate LSP databases for Level-1 LSPs and Level-2 LSPs.

Displaying summary information

To display summary information for all the LSPs in the PowerConnect's LSP databases, enter the following command at any level of the CLI.

```
NetIron)# show isis database
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
MLXe-1.00-00           0x0000000c   0xd048        963               1/0/0
MLXe-1.01-00           0x00000004   0x09b0        957               0/0/0
MLXe-1.02-00           0x00000001   0xc57b        961               0/0/0
MLXe.00-00*            0x0000000b   0x23fb        1030              1/0/0

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
MLXe-1.00-00           0x0000000d   0x7d97        964               1/0/0
MLXe-1.01-00           0x00000004   0x09b0        958               0/0/0
MLXe-1.02-00           0x00000001   0x200f        962               0/0/0
MLXe.00-00*            0x0000000b   0x5647        1030              1/0/0
0000.0100.0003.00-00  0x0000001f   0x761a        932               0/0/0
0000.0100.0003.00-01  0x0000001d   0x9c9d        606               0/0/0
```

The command in this example shows information for the LSPS in the PowerConnect's Level-1 and Level-2 LSP databases. Notice that the display groups the Level-1 and Level-2 LSPs separately.

Syntax: show isis database [<*lsp-id*> | **detail** | **l1** | **l2** | **level1** | **level2**]

The <*lsp-id*> parameter displays summary information about a particular LSP. Specify an LSPID for which you want to display information in HHHH.HHHH.HHHH.HH-HH format, for example, 3333.3333.3333.00-00. You can also enter name.HH-HH, for example, MLXe.00-00.

The **detail** parameter displays detailed information about the LSPs. Refer to "Displaying detailed information" on page 986.

The **l1** and **level1** parameters display the Level-1 LSPs only. You can use either parameter.

The **l2** and **level2** parameters display the Level-2 LSPs only. You can use either parameter.

The **show isis database** summary display shows the following information.

TABLE 159 IS-IS summary LSP database information

This field...	Displays...
LSPID	The LSP ID, which consists of the source ID (6 bytes), the pseudonode (1 byte), and LSPID (1 byte). **NOTE:** If the address has an asterisk (*) at the end, this indicates that the LSP is locally originated.
LSP Seq Num	The sequence number of the LSP.
LSP Checksum	The checksum calculated by the device that sent the LSP and used by the PowerConnect to verify that the LSP was not corrupted during transmission over the network.
LSP Holdtime	The maximum number of seconds during which the LSP will remain valid. **NOTE:** The IS that originates the LSP sets the timer for the LSP. As a result, LSPs do not all have the same amount of time remaining when they enter the PowerConnect's LSP database.
ATT	A 4-bit value extracted from bits 4 – 7 in the Attach field of the LSP.
P	The value in the Partition option field of the LSP. The field can have one of the following values: • 0 – The IS that sent the LSP does not support partition repair. • 1 – The IS that sent the LSP supports partition repair.
OL	The value in the LSP database overload field of the LSP. The field can have one of the following values: • 0 – The overload bit is off. • 1 – The overload bit is on, indicating that the IS that sent the LSP is overloaded and should not be used as a IS-IS transit router for that level.

Displaying detailed information

To display detailed information for all the LSPs in the PowerConnect's LSP databases, enter the following command at any level of the CLI.

```
NetIron# show isis database detail
IS-IS Level-1 Link State Database
LSPID  LSP Seq Num LSP Checksum      LSP Holdtime ATT/P/OL
MLX.00-00*    0x0000000b      0x23fb               971       1/0/0

Area Address:   49
NLPID:  CC(IP)
Hostname:  MLX14
IP Address: 17.1.1.1
IPv6 Address: 2001::14
Metric: 10      IP-Internal       4.1.1.0/24             Up-bit: 0
Metric:  10     IS MLX.01

IS-IS Level-2 Link State Database
LSPID                    LSP Seq Num    LSP Checksum         LSP Holdtime
MLX.00-00*         0x0000000d        0x7d97                    903
1/0/0

Area Address:  49
NLPID:   IPv6IP
Hostname:   MLX14
IP address:   4.1.1.1
IPv6 address: 2001::14
Flooding to 1 interface: eth 1/7
Acking to 1 interface: pos 4/1
Metric: 10     IP-Internal           4.1.1.0/24           Up-bit: 0
Metric: 10     IP-Internal           192.85.1.0/24    Up-bit: 0
Metric: 10     IS MLX.01
Metric: 10     IS MLX.02
```

TABLE 160 IS-IS detailed LSP database information

This field...	Displays...
LSPID	Refer to the description of the summary display.
LSP Seq Num	Refer to the description of the summary display.
LSP Checksum	Refer to the description of the summary display.
LSP Holdtime	Refer to the description of the summary display.
ATT or P or OL	Refer to the description of the summary display.
Area Address	The address of the area.
NLPID	The Network Layer Protocol Identifier (NLPID), which specifies the protocol the IS that sent the LSP is using. Usually, this value is "CC(IP)".
IP address	The IP address of the interface that sent the LSP. The PowerConnect can use this address as the next hop in routes to the addresses listed in the rows below.

TABLE 160 IS-IS detailed LSP database information (Continued)

This field...	Displays...

Destination addresses	The rows of information below the IP address row are the destinations advertised by the LSP. The PowerConnect can reach these destinations by using the IP address listed above as the next hop.
	Each destination entry contains the following information:
	• Metric – The value of the default metric, which is the IS-IS cost of using the IP address above as the next hop to reach this destination.
	• Device type – The device type at the destination. The type can be one of the following:
	  • End System – The device is an ES.
	  • IP-Internal – The device is an ES within the current area. The IP address and subnet mask are listed.
	  • IS – The device is another IS. The NET (NSAP address) is listed.
	  • IP-Extended – Same as IP-Internal, except the device uses the extended TLV fields described in draft-ietf-isis-traffic-02.txt to carry the information.
	  • IS-Extended – Same as IS, except the device uses the extended TLV fields described in draft-ietf-isis-traffic-02.txt to carry the information.

Flooding to <*num*> interface:	Identifies the number of interfaces on which the specific LSP entry will be flooded and identifies the interfaces.
Acking to <*num*> interface:	Identifies the number of interfaces on which the specific LSP entry will be acknowledged and identifies the interfaces.

Displaying database summary information

The following command is used to display the ISIS database.

```
NetIron# show isis database summary
IS-IS Level-1 Link State Database Summary
Number of LSPs :                    2
Number of LSPs loading :            0
Number of LSP fragments :           0
Number of Pseudo LSPs :             1
Number of Pseudo LSP fragments :    0
Number of My LSPs :                 1
Number of My LSP fragments :        0
Number of My Pseudo LSPs :          0
Number of My Pseudo LSP fragments : 0
Sum of LSPs Checksum :              0x00018004

IS-IS Level-2 Link State Database Summary

Number of LSPs :                    2
Number of LSPs loading :            0
Number of LSP fragments :           0
Number of Pseudo LSPs :             1
Number of Pseudo LSP fragments :    0
Number of My LSPs :                 1
Number of My LSP fragments :        0
```

```
Number of My Pseudo LSPs :          0
Number of My Pseudo LSP fragments : 0
Sum of LSPs Checksum :              0x00019775
```

Table 161 defines the fields shown in the above example output of the **show ip ospf interface brief** command.

TABLE 161 Output of the **show isis database summary** command

This field	Displays
Number of LSPs	Total number of LSPs in database (includes those in the loading state).
Number of LSPs loading	Number of LSPs pending a full LSP update. This value is generally non-zero during adjacency formation.
Number of LSP fragments	The number of LSPs with a non-zero LSP number (a fragment of an LSP)
Number of Pseudo LSPs	The number of pseudo LSPs.
Number of Pseudo LSP fragments	The number of pseudo LSPs with a non-zero LSP number (a fragment of an LSP).
Number of My LSPs	Total number of LSPs originated by this router.
Number of My LSP fragments	The number of LSPs originated by this router with a non-zero LSP number (a fragment of an LSP)
Number of My Pseudo LSPs	The number of pseudo LSPs originated by this router.
Number of My Pseudo LSP fragments	The number of pseudo LSPs originated by this router with a non-zero LSP number (a fragment of an LSP).
Sum of LSPs Checksum	Total checksum of all LSPs in database (including those in loading state). This number should be the same across ISIS routers during periods of network stability.

Displaying traffic statistics

The PowerConnect maintains statistics for common IS-IS PDU types. To display the statistics, use either of the following methods.

To display IS-IS PDU statistics, enter the following command at any level of the CLI.

```
NetIron# show isis traffic
                            Message Received     Message Sent
 Level-1 Hellos             1029                 115
 Level-2 Hellos             1027                 112
 Level-1 LSP                6                    3
 Level-2 LSP                6                    3
 Level-1 CSNP               0                    0
 Level-2 CSNP               0                    0
 Level-1 PSNP               107                  0
 Level-2 PSNP               107                  0
```

Syntax: show isis traffic

This display shows the following information.

TABLE 162 IS-IS traffic statistics

This field...	Displays...
Level-1 Hellos	The number of Level-1 hello PDUs sent and received by the PowerConnect.
Level-2 Hellos	The number of Level-2 hello PDUs sent and received by the PowerConnect.
Level-1 LSP	The number of Level-1 link-state PDUs sent and received by the PowerConnect.
Level-2 LSP	The number of Level-2 link-state PDUs sent and received by the PowerConnect.
Level-1 CSNP	The number of Level-1 Complete Sequence Number PDUs (CSNPs) sent and received by the PowerConnect.
Level-2 CSNP	The number of Level-2 CSNPs sent and received by the PowerConnect.
Level-1 PSNP	The number of Level-1 Partial Sequence Number PDUs (PSNPs) sent and received by the PowerConnect.
Level-2 PSNP	The number of Level-2 PSNPs sent and received by the PowerConnect.

Displaying error statistics

To display IS-IS error statistics, enter the following command at any level of the CLI.

```
NetIron# show isis counts
Area Mismatch: 0
Max Area Mismatch: 0
System ID Length Mismatch: 0
LSP Sequence Number Skipped: 0
LSP Max Sequence Number Exceeded: 0
Level-1 Database Overload: 0
Level-2 Database Overload: 0
Our LSP Purged: 0
PDU Drop Count
CSNP Auth Failures : [L1: 100] [L2: 0]
PSNP Auth Failures : [L1: 100] [L2: 0]
HELLO Auth Failures : [L1: 100] [L2: 0]
Adjacency not found : [L1: 100] [L2: 200]
Adjacency Level Mismatch : [L1: 100] [L2: 200]
IS Level Mismatch : [L1: 100] [L2: 200]
Length Too Short : [L1: 100] [L2: 200]
Length Too Large : [L1: 100] [L2: 200]
Max Area Check Failure : [L1: 100] [L2: 200]
Zero Checksum : [L1: 100] [L2: 200]
Checksum Mismatch : [L1: 100] [L2: 200]
Invalid Length : [L1: 100] [L2: 200]
```

Syntax: show isis counts

This display shows the following information.

TABLE 163 IS-IS error statistics

This field...	Displays...
Area Mismatch	The number of times the PowerConnect interface was unable to create a Level-1 adjacency with a neighbor because the PowerConnect interface and the neighbor did not have any areas in common.
Max Area Mismatch	The number of times the PowerConnect received a PDU whose value for maximum number of area addresses did not match the PowerConnect's value for maximum number of area addresses.
System ID Length Mismatch	The number of times the PowerConnect received a PDU whose ID field was a different length than the ID field length configured on the PowerConnect.
LSP Sequence Number Skipped	The number of times the PowerConnect received an LSP with a sequence number that was more than 1 higher than the sequence number of the previous LSP received from the same neighbor.
LSP Max Sequence Number Exceeded	The number of times the PowerConnect attempted to set an LSP sequence number to a value higher than the highest number in the CSNP sent by the Designated IS.
Level-1 Database Overload	The number of times the Level-1 state on the PowerConnect changed from Waiting to On or from On to Waiting. • Waiting to On – This change can occur when the PowerConnect recovers from a previous Level-1 LSP database overload and is again ready to receive new LSPs. • On to Waiting – This change can occur when the PowerConnect's Level-1 LSP database is full and the PowerConnect receives an additional LSP, for which there is no room.
Level-2 Database Overload	The number of times the Level-2 state on the PowerConnect changed from Waiting to On or from On to Waiting. • The change from Waiting to On can occur when the PowerConnect recovers from a previous Level-2 LSP database overload and is again ready to receive new LSPs. • The change from On to Waiting can occur when the PowerConnect's Level-2 LSP database is full and the PowerConnect receives an additional LSP, for which there is no room.
Our LSP Purged	The number of times the PowerConnect received an LSP that was originated by the PowerConnect itself and had age zero (aged out).
PDU Drop Count	
CSNP Auth Failures	The number of CSNP Authentication failures recorded for Level-1 and Level-2. This counter will only be displayed if it has a value greater than zero.
PSNP Auth Failures	The number of PSNP Authentication failures recorded for Level-1 and Level-2. This counter appears only if it has a value greater than 0.
HELLO Auth Failures	The number of HELLO Authentication failures recorded for Level-1 and Level-2. This counter will only be displayed if it has a value greater than zero.
Adjacency not found	The number of PDUs dropped at both Level-1 and Level-2 because there is no valid adjacency on the interface where they were received. This counter will only be displayed if it has a value greater than zero.

TABLE 163 IS-IS error statistics (Continued)

This field...	Displays...
Adjacency Level Mismatch	The number of PDUs dropped at both Level-1 and Level-2 because the adjacency from which the PDU is received has a different level than the PDU level. This counter will only be displayed if it has a value greater than zero.
IS Level Mismatch	The number of PDUs dropped at both Level-1 and Level-2 because the IS-IS router level mismatches with the PDU level received. This counter will only be displayed if it has a value greater than zero.
Length Too Short	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU length is less than the standard PDU header length. This counter will only be displayed if it has a value greater than zero.
Length Too Long	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU length is greater than the MTU of the link. This counter will only be displayed if it has a value greater than zero.
Max Area Check Failure	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU has a maximum area count different than what is configured on this IS-IS router. This counter will only be displayed if it has a value greater than zero.
Zero Checksum	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU has a zero checksum. This counter will only be displayed if it has a value greater than zero.
Checksum Mismatch	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU has a checksum different than the computed checksum on the received PDU. This counter will only be displayed if it has a value greater than zero.
Invalid Length	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU has a different length than what is advertised in the PDU header. This counter will only be displayed if it has a value greater than zero.

Displaying the IS-IS SPF Log

The **show isis spf-log** command displays the ISIS Log, as shown in the following.

```
NetIron#show isis spf-log detail
ISIS Level-1 SPF Log
When      Duration  Nodes  Count  Last-Trigger-LSP         Trigger
0h1m57s   0         3      2      mu1.00-00                Adjacency Change
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
     First Trigger: 0h1m45s       Adj TLV Changed in LSP mu2.00-00
     Last Trigger : 0h1m45s       Adj TLV Changed in LSP mu1.00-00
0h2m3s    0         3      2      mu2.00-00                New LSP
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
     First Trigger: 1h42m45s      Adjacency mu2 is added
     Last Trigger : 1h42m45s      New LSP mu2.00-00 Appeared in database
0h2m9s    0         0      3      mu1.00-00                New LSP
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
     First Trigger: 1h42m45s      Interface ve 3 is Up
     Last Trigger : 1h42m45s      New LSP mu1.00-00 Appeared in database

ISIS Level-2 SPF Log
When      Duration  Nodes  Count  Last-Trigger-LSP          Trigger
0h2m9s    0         0      3      mu1.00-00                 New LSP
 Ipv4 Route updates: 4000         Ipv6 Route updates: 0
     First Trigger: 1h42m45s      Interface ve 3 is Up
     Last Trigger : 1h42m45s      New LSP mu1.00-00 Appeared in database
0h2m21s   0         0      6      mu1.00-00                 New LSP
 Ipv4 Route updates: 4000         Ipv6 Route updates: 0
     First Trigger: 1h42m45s      Interface eth 1/1 is Up
     Last Trigger : 1h42m45s      New LSP mu1.00-00 Appeared in database
0h3m21s   0         0      3      mu1.00-00                 Adjacency Change
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
     First Trigger: 1h42m45s      New LSP mu1.00-00 Appeared in database
     Last Trigger : 1h42m45s      Adj TLV is Changed in LSP mu1.00-00
```

Syntax: show isis spf-log {detail | level-1 [detail] | level-2 [detail] }

This display shows the following information.

TABLE 164 IS-IS SPF log information

This field...	Displays...
When	When (in hours: minutes : seconds) a full SPF calculation occurred. The last 20 occurrences are logged.
Duration	The time required to complete this SPF run, Elapsed time is normal clock time (not CPU time). Other options for this field are: • Running – the SPF is still running and the duration will be updated after the SFP has run. • Pending – the event is pending and another SPF will be run once the currently executing SPF has completed.
Nodes	The number of routers and pseudonodes (LANs) that make up the topology calculated in this SPF run.
Count	The number of events that triggered this SPF run. When a topology change has occurred, multiple link-state packets (LSPs) are received in a short time. Since a router waits about 5 seconds before running a full SPF run, it can include all new information. This count includes the number of events (such as receiving new LSPs) that occurred while the router was waiting the 5 second interval before running full SPF.

TABLE 164 IS-IS SPF log information (Continued)

This field...	Displays...
Last Trigger LSP	When a full SPF calculation is triggered by the arrival of a new LSP, the router stores the LSP ID. The LSP ID can provide a clue about the source of routing instability in an area. If multiple LSPs in a single level are causing SPF runs, only the LSP ID of the last received LSP is recorded.
Triggers	The reason that a full SPF calculations was triggered. Table 165 describes each of the triggers that can be displayed in this field.

For a description of the trigger types, refer to the following table.

TABLE 165 Trigger types and description

Trigger	Description
Alternate Route Check	PSPF deleted an IPv4 or IPv6 route. Full SPF must run to find the alternate route.
Route Change in L1 SPF Run	The L1 SPF run added or deleted an IPv4 or IPv6 route. The L2 SPF must run to accommodate this change.
LSP Purged	An LSP was purged. A full SPF calculation must process this change.
LSP Added	A new LSP has appeared in the database. A full SPF calculation is needed to process this new LSP.
Summary Address Change	A summary address configuration change has occurred.
Adjacency State Change	An adjacency was added or deleted.
Admin Distance Change	The administrative distance configuration has changed.
LSP Header Change	The LSP header (attached or overload bits) is changed.
IS Neighbor TLV Change	An IS neighbor TLV was added or deleted in an LSP.
Area Address TLV Change	The area address TLV changed.
Interface IP Address Change	The IP address configuration changed.
IP Address TLV Change	An IP address TLV changed in the LSP.
IPv6 Address TLV Change	An IPv6 address TLV changed in the LSP.
IS-IS Level Change	The IS-IS level configuration changed.
Interface Metric Change	The IS-IS interface metric configuration changed.
LSP Changed - PSPF Disabled	The LSP changed and PSPF is disabled.
LSP Overload Bit Change	The overload bit in the LSP header changed.
Interface State Change	The interface state changed to up or down.
Redist Prefix-List Change	The redistribution list configuration changed.
Redist Policy Change	The redistribution policy configuration changed.
Maximum Path Change	The IS-IS maximum path configuration changed.
IP Load Sharing Change	The IP load sharing configuration changed.
User Cleared IS-IS Route	The user cleared a specific IS-IS route.
User Cleared IS-IS Routes	The user cleared all IS-IS routes.
Neighbor NLPID Change	NLPID set is changed in received hellos.
ISIS Enable	IS-IS was enabled.

TABLE 165 Trigger types and description (Continued)

Trigger	Description
User Cleared IS-IS All	The user issued the **clear isis all** command.
Interface Config Change	ISIS was enabled or disabled on a port.
User Trigger	The user issued the **clear isis spf-trigger** command.
Recompute InterLeve Routes	The neighbor IS-type is changed either from L1 to L12 or L12 to L1
Exited Overload State	IS-IS exited from an overload condition.

By using the **detail** option with the **show isis spf-log** command, you can display more detail about the total number of IPv4 and IPv6 route updates and the reason for the first and last SPF events. Like SPF events, the incremental SPF events are displayed. However, for incremental SPF, only the first trigger is displayed, as the example below illustrates. In addition, the logging changes include the number of RTM updates that were carried out in each SPF or incremental SPF run.

To show details about the RTM updates, use the **show isis spf-log detail** command, as follows.

```
NetIron#show isis spf-log detail
ISIS Level-2 SPF Log
 When     Duration Nodes Count Last-Trigger-LSP     Trigger

2h38m9s   0ms      3     2     MLXe14.00-00          Adjacency State Change
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
       First Trigger: 2h39m23s loopback 1 State Changed to Up
       Last Trigger : 2h38m14s Adjacency 0000.1100.0001 Added
2h41m17s 0ms       26    1     MLXe14.00-00          IS Neighbor TLV Change
 Ipv4 Route updates: 1 Ipv6 Route updates: 0
       First Trigger: 2h41m2s ISPF Run
```

Syntax: show isis spf-log detail

Clearing the IS-IS SPF Log

You can clear the IS-IS SPF Log accumulated since the last software reload or last clearing of the SPF Log through use of the following command.

```
NetIron# isis clear spf-log
```

Syntax: clear isis spf-log [level-1 | level-2]

When the **level-1** or **level-2** options are used, only the log for the specified level is cleared. If not specified, both will be cleared.

Triggering the router to run SPF

You can trigger the router to run the SPF calculations through use of the following command.

```
NetIron# clear isis spf-trigger
```

Syntax: clear isis spf-trigger [level-1 | level-2]

When the **level-1** or **level-2** options are used, the SPF calculation is only triggered for the specified level. If not specified, the SPF calculation will be triggered for both.

Clearing IS-IS information

To clear the IS-IS information that the PowerConnect has accumulated since the last time you cleared information or reloaded the software, use either of the following methods.

To clear IS-IS information, enter the **clear isis all** command at any level of the CLI except the User EXEC level.

```
NetIron# clear isis all
```

This command clears all the following:

- Neighbors (closes the PowerConnect's adjacencies with its IS-IS neighbors)
- Routes
- PDU statistics
- Error statistics

Syntax: **clear isis all | counts | neighbor | route** [<*ip-address*> <*subnet-mask*> **|** <*ip-address*>/<*prefix*>] **| traffic**

The **all** parameter clears all the IS-IS information. Using this option is equivalent to entering separate commands with each of the other options.

The **counts** parameter clears the error statistics.

The **neighbor** parameter closes the PowerConnect's adjacencies with its IS-IS neighbors and clears neighbor statistics.

The **route** [<*ip-address*> <*subnet-mask*> | <*ip-address*>/<*prefix*>] parameter clears the IS-IS route table or the specified matching route.

The **traffic** parameter clears the PDU statistics.

NOTE
The **traffic** option also clears the values displayed in the **show isis interface** command's Control Messages Sent and Control Messages Received fields.

The **neighbor** option of the **clear isis** command has been enhanced as described in the following:

Syntax: **clear isis neighbor all [ethernet** <*slot/port*> **| pos** <*slot/port*> **| tunnel** <*tunnel-id*> **| ve** <*port-number*>]

The **all** option directs the router to clear all neighbors on all IS-IS interfaces or clear all neighbors on an interface specified using one of the following options:

ethernet <*slot/port*> – clears all IS-IS neighbors on the specified Ethernet interface.

pos <*slot/port*> – clears all IS-IS neighbors on the specified POS interface.

ve <*port-no*> – clears all IS-IS neighbors on the specified virtual interface.

tunnel <*tunnel-port*> – clears all IS-IS neighbors on the specified tunnel interface.

Syntax: **clear isis neighbor** <*sys-id*> **[ethernet** <*slot/port*> **| pos** <*slot/port*> **| tunnel** <*tunnel-id*> **| ve** <*port-number*>]

This command directs the router to clear the IS-IS neighbor specified by the <*sys-id*> variable on all possible interfaces or to clear the IS-IS neighbor specified by the <*sys-id*> variable on an interface specified using one of the following options:

ethernet <*slot/port*> – clears the specified IS-IS neighbor on the specified Ethernet interface.

pos <*slot/port*> – clears the specified IS-IS neighbor on the specified POS interface.

ve <*port-no*> – clears the specified IS-IS neighbor on the specified virtual interface.

tunnel <*tunnel-port*> – clears the specified IS-IS neighbor on the specified tunnel interface.

Clearing a specified LSP from IS-IS database

A new command has been added that allows you to clear a specified LSP from the IS-IS database. Running this command causes the regeneration of the specified LSP where this LSP was originated by this router. For example, to clear the LSP named "MLXe-1.00-00" from the IS-IS database, enter the following command.

```
NetIron# clear isis database MLXe-1.00-00
```

Syntax: clear isis database <*lsp-id*> **[level-1 | level-2 | level-1-2]**

The <*lsp-id*> parameter displays summary information about a particular LSP. Specify an LSPID for which you want to display information in HHHH.HHHH.HHHH.HH-HH format, for example, 3333.3333.3333.00-00. You can also enter name.HH-HH, for example, MLXe.00-00.

The **level-1**parameter limits you to clear level-1 LSPs only.

The **level-2** parameter limits you to clear level-2 LSPs only.

The **level-1-2** parameter clears level-1 and level-2 LSPs. This is the default.

NOTE
The **clear isis all** command should be used to regenerate the complete database.

Configuring BGP4 (IPv4) 26

Overview

The following displays the BGP4 features supported by PowerConnect B-MLXe.

- BGP4
- BGP4 Restart
- BGP4 Restart Helper Mode
- Redistributing IBGP Routes
- Client-to-Client Routes
- Route Flap Dampening
- Originating the Default Route
- Multipath Load Sharing
- Using the IP Default Route as a Valid Next Hop for a BGP4 Route
- Next-Hop Recursion
- Next-Hop Update Timer
- Generalized TTL Security Mechanism Support
- Enhanced per-neighbor debug statements and new per-neighbor BGP4 debug filters
- BGP4 Peer Notification During a Management Module Switchover
- Auto Shutdown of BGP4 Neighbors on Initial Configuration
- New encryption code for passwords, authentication keys, and community strings
- BGP4 MD5 Authentication
- Route redistribution to other protocols
- BGP4 Peer Group
- BGP4 Route Reflectors
- BGP4 Neighbor Local-AS
- BGP4 Processing Optimization for Administratively Down Peers
- BGP4 Outbound Policy Processing Optimization
- Generalized TTL Security Mechanism Support
- Enhanced per-neighbor debug statements and new per-neighbor BGP4 debug filters
- Requiring the First AS to be the Neighbor's AS
- Four-byte AS Numbers (AS4)
- BGP4 AS4 Confederation Error Checking
- RTM Scalability Enhancement
- Route Map Continue Clause
- Static BGP4 Networks

- Limiting Advertisement of a Static BGP4 Network
- Use IGP cost instead of BGP MED value

This chapter provides details on how to configure Border Gateway Protocol version 4 (BGP4).

NOTE
BGP4 commands that are supported in IPv4 are listed in Table 166.

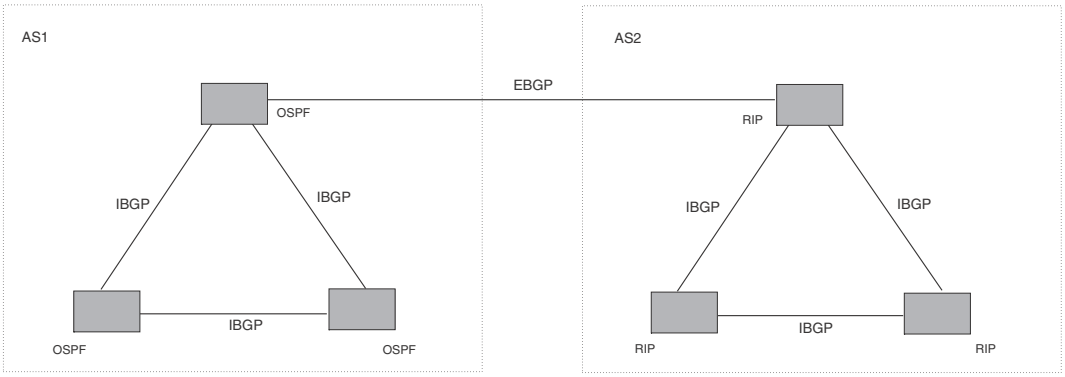
Overview of BGP4

BGP4 is the standard Exterior Gateway Protocol (EGP) used on the Internet to route traffic between Autonomous Systems (AS) and to maintain loop-free routing. An autonomous system is a collection of networks that share the same routing and administration characteristics. For example, a corporate Intranet consisting of several networks under common administrative control might be considered an AS. The networks in an AS can but do not need to run the same routing protocol to be in the same AS, nor do they need to be geographically close.

Devices within an AS can use different Interior Gateway Protocols (IGPs) such as RIP and OSPF to communicate with one another. However, for devices in different ASs to communicate, they need to use an EGP. BGP4 is the standard EGP used by Internet devices and therefore is the EGP implemented on the device.

Figure 146 shows a simple example of two BGP4 ASs. Each AS contains three BGP4 devices. All of the BGP4 devices within an AS communicate using IBGP. BGP4 devices communicate with other ASs using EBGP. Notice that each of the devices also is running an Interior Gateway Protocol (IGP). The devices in AS1 are running OSPF and the devices in AS2 are running RIP. The device can be configured to redistribute routes among BGP4, ISIS, RIP, and OSPF. They also can redistribute static routes.

FIGURE 146 Example BGP4 ASs



Relationship between the BGP4 route table and the IP route table

The device BGP4 route table can have multiple routes or paths to the same destination, which are learned from different BGP4 neighbors. A BGP4 neighbor is another device that also is running BGP4. BGP4 neighbors communicate using Transmission Control Protocol (TCP) port 179 for BGP4 communication. When you configure the device for BGP4, one of the configuration tasks you perform is to identify the device's BGP4 neighbors.

Although a device's BGP4 route table can have multiple routes to the same destination, the BGP4 protocol evaluates the routes and chooses only one of the routes to send to the IP route table. The route that BGP4 chooses and sends to the IP route table is the ***preferred route.*** This route is what the device advertises to other BGP4 neighbors. If the preferred route goes down, BGP4 updates the route information in the IP route table with a new BGP4 preferred route.

NOTE
If IP load sharing is enabled and you enable multiple equal-cost paths for BGP4, BGP4 can select more than one equal-cost path to a destination.

A BGP4 route consists of the following information:

- **Network number (prefix)** – A value comprised of the network mask bits and an IP address (*<IP address>/ <mask bits>*); for example, 192.215.129.0/18 indicates a network mask of 18 bits applied to the IP address 192.215.129.0. When a BGP4 device advertises a route to one of its neighbors, it uses this format.
- **AS-path** – A list of the other ASs through which a route passes. BGP4 devices can use the AS-path to detect and eliminate routing loops. For example, if a route received by a BGP4 device contains the AS that the device is in, the device does not add the route to its own BGP4 table. (The BGP4 RFCs refer to the AS-path as "AS_PATH", and RFC 4893 uses "AS4_PATH" in relation to AS4s.)
- **Additional path attributes** – A list of additional parameters that describe the route. The route MED and next hop are examples of these additional path attributes.

NOTE
The device re-advertises a learned best BGP4 route to the device's neighbors even when the route table manager does not select that route for installation in the IP route table. This can happen if a route from another protocol, for example, OSPF, is preferred. The best BGP4 route is the route that BGP4 selects based on comparison of the BGP4 route path's attributes.

After a device successfully negotiates a BGP4 session with a neighbor (a BGP4 peer), the device exchanges complete BGP4 route tables with the neighbor. After this initial exchange, the device and all other RFC 1771-compliant BGP4 devices send UPDATE messages to inform neighbors of new, changed, or no longer feasible routes. BGP4 devices do not send regular updates. However, if configured to do so, a BGP4 device does regularly send KEEPALIVE messages to its peers to maintain BGP4 sessions with them if the device does not have any route information to send in an UPDATE message. Refer to "BGP4 message types" on page 1001 for information about BGP4 messages.

How BGP4 selects a path for a route

When multiple paths for the same route prefix are known to a BGP4 device, the device uses the following algorithm to weigh the paths and determine the optimal path for the route. The optimal path depends on various parameters, which can be modified.

1. Is the next hop accessible though an Interior Gateway Protocol (IGP) route? If not, ignore the route.

NOTE
By default, the device does not use the default route to resolve BGP4 next hop. Refer to "Enabling next-hop recursion" on page 1051 and "Using the IP default route as a valid next-hop for a BGP4 route" on page 1051.

2. Use the path with the largest weight.
3. If the weights are the same, prefer the path with the largest local preference.
4. Prefer the route that was originated locally (by this device).
5. If the local preferences are the same, prefer the path with the shortest AS-path. An AS-SET counts as 1. A confederation path length, if present, is not counted as part of the path length.

NOTE
This step can be skipped if **BGP4-as-path-ignore** is configured.

6. If the AS-path lengths are the same, prefer the path with the lowest origin type. From low to high, route origin types are valued as follows:
   - IGP is lowest.
   - EGP is higher than IGP but lower than INCOMPLETE.
   - INCOMPLETE is highest.
7. If the paths have the same origin type, prefer the path with the lowest MED. For a definition of MED, refer to "Configuring the device to always compare Multi-Exit Discriminators" on page 1014".
   - device compares the MEDs of two otherwise equivalent paths ***if and only if*** the routes were learned from the same neighboring AS. This behavior is called ***deterministic MED***. Deterministic MED is always enabled and cannot be disabled.

     You can also enable the device to always compare the MEDs, regardless of the AS information in the paths. To enable this comparison, enter the **always-compare-med** command at the BGP4 configuration level of the CLI. This option is disabled by default.

NOTE
By default, value 0 (most favorable) is used in MED comparison when the MED attribute is not present. The default MED comparison results in the device favoring the route paths that are missing their MEDs. You can use the **med-missing-as-worst** command to make the device regard a BGP4 route with a missing MED attribute as the least favorable path, when comparing the MEDs of the route paths.

NOTE
MED comparison is not performed for internal routes originated within the local AS or confederation unless the **compare-med-empty-aspath** command is configured.

8. Prefer paths in the following order:
    - Routes received through EBGP from a BGP4 neighbor outside of the confederation
    - Routes received through EBGP from a BGP4 device within the confederation
    - Routes received through IBGP
9. If all the comparisons above are equal, prefer the route with the lowest IGP metric to the BGP4 next hop. This is the closest internal path inside the AS to reach the destination.
10. If the internal paths also are the same and BGP4 load sharing is enabled, load share among the paths. Otherwise go to step 11.

NOTE
The device supports BGP4 load sharing among multiple equal-cost paths. BGP4 load sharing enables the device to balance traffic across the multiple paths instead of choosing just one path based on device ID. For EBGP routes, load sharing applies only when the paths are from neighbors within the same remote AS. EBGP paths from neighbors in different ASs are not compared, unless multipath **multi-as** is enabled.

11. If **compare-router ID** is enabled, prefer the path that comes from the BGP4 device with the lowest device ID. If a path contains originator ID attributes, then originator ID is substituted for the ROUTER ID in the decision.
12. Prefer the path with the minimum cluster list length.
13. If the route is a BGP4 VRF instance, prefer the route with the smallest RD value.
14. Prefer the route that comes from the lowest BGP4 neighbor address.

BGP4 message types

BGP4 devices communicate with neighbors (other BGP4 devices) using the following types of messages:

- OPEN
- UPDATE
- KEEPALIVE
- NOTIFICATION
- ROUTE REFRESH

OPEN message

After a BGP4 device establishes a TCP connection with a neighboring BGP4 device, the devices exchange OPEN messages. An open message indicates the following:

- **BGP4 version** – Indicates the version of the protocol that is in use on the device. BGP4 version 4 supports Classless Interdomain Routing (CIDR) and is the version most widely used in the Internet. Version 4 also is the only version supported on the device.
- **AS number** – An autonomous system number (ASN) identifies the AS to which the BGP4 device belongs. The number can be up to four bytes.

Hold Time – The number of seconds a BGP4 device will wait for an UPDATE or KEEPALIVE message (described below) from a BGP4 neighbor before assuming that the neighbor is not operational. BGP4 devices exchange UPDATE and KEEPALIVE messages to update route information and maintain communication. If BGP4 neighbors are using different Hold Times, the lowest Hold Time is used by the neighbors. If the Hold Time expires, the BGP4 device closes the TCP connection to the neighbor and clears any information it has learned and cached from the neighbor.

You can configure the Hold Time to be 0, in which case a BGP4 device will consider neighbors to always be up. For directly-attached neighbors, you can configure the device to immediately close the TCP connection to the neighbor and clear entries learned from an EBGP neighbor if the interface to that neighbor goes down. This capability is provided by the fast external fail over feature, which is disabled by default.

- **BGP4 Identifier** – The device ID. The BGP4 Identifier (device ID) identifies the BGP4 device to other BGP4 devices. The device use the same device ID for OSPF and BGP4. If you do not set a device ID, the software uses the IP address on the lowest numbered loopback interface configured on the device. If the device does not have a loopback interface, the default device ID is the lowest numbered IP address configured on the device. For more information, or to change the device ID, refer to "Changing the router ID" on page 684.
- **Parameter list** – An optional list of additional parameters used in peer negotiation with BGP4 neighbors.

UPDATE message

After BGP4 neighbors establish a BGP4 connection over TCP and exchange their BGP4 routing tables, they do not send periodic routing updates. Instead, a BGP4 neighbor sends an update to a neighbor when it has a new route to advertise or routes have changed or become unfeasible. An UPDATE message can contain the following information:

- **Network Layer Reachability Information (NLRI)** – The mechanism by which BGP4 supports Classless Interdomain Routing (CIDR). An NLRI entry consists of an IP prefix that indicates a network being advertised by the UPDATE message. The prefix consists of an IP network number and the length of the network portion of the number. For example, an UPDATE message with the NLRI entry 192.215.129.0/18 indicates a route to IP network 192.215.129.0 with network mask 255.255.192.0. The binary equivalent of this mask is 18 consecutive one bits, thus "18" in the NLRI entry.
- **Path attributes** – Parameters that indicate route-specific information such as AS path information, route preference, next hop values, and aggregation information. BGP4 uses path attributes to make filtering and routing decisions.
- **Unreachable routes** – A list of routes that have been in the sending device BGP4 table but are no longer feasible. The UPDATE message lists unreachable routes in the same format as new routes: *<IP address>/<CIDR prefix>*.

KEEPALIVE message

BGP4 devices do not regularly exchange UPDATE messages to maintain BGP4 sessions. For example, if a device configured to perform BGP4 routing has already sent the latest route information to peers in UPDATE messages, the device does not send more UPDATE messages. Instead, BGP4 devices send KEEPALIVE messages to maintain BGP4 sessions. KEEPALIVE messages are 19 bytes long and consist only of a message header. They do not contain routing data.

BGP4 devices send KEEPALIVE messages at a regular interval, called the Keep Alive Time. The default Keep Alive Time is 60 seconds.

A parameter related to the Keep Alive Time is the Hold Time. The Hold Time for a BGP4 device determines how many seconds the device waits for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is not operational. The Hold Time is negotiated when BGP4 devices exchange OPEN messages, the lower Hold Time is then used by both neighbors. For example, if BGP4 device A sends a Hold Time of 5 seconds and BGP4 device B sends a Hold Time of 4 seconds, both devices use 4 seconds as the Hold Time for their BGP4 session. The default Hold Time is 180 seconds. Generally, the Hold Time is configured to three times the value of the Keep Alive Time.

If the Hold Time is 0, a BGP4 device assumes that a neighbor is alive regardless of how many seconds pass between receipt of UPDATE or KEEPALIVE messages.

NOTIFICATION message

When you close the BGP4 session with a neighbor, the device detects an error in a message received from the neighbor, or an error occurs on the device, the device sends a NOTIFICATION message to the neighbor. No further communication takes place between the BGP4 device that sent the NOTIFICATION and the neighbors that received the NOTIFICATION.

REFRESH message

BGP4 sends a REFRESH message to a neighbor to request that the neighbor resend route updates. This type of message can be useful if an inbound route filtering policy has been changed.

Implementation of BGP4

BGP4 is described in RFC 1771 and the latest BGP4 drafts. The implementation fully complies with RFC 1771 and supports the following:

- RFC 1745 (OSPF Interactions)
- RFC 1997 (BGP4 Communities Attributes)
- RFC 2385 (TCP MD5 Signature Option)
- RFC 2439 (Route Flap Dampening)
- RFC 4456 (Route Reflection)
- RFC 2842 and 3392 (Capability Advertisement)
- RFC 3065 (BGP4 Confederations)
- RFC 2858 (Multiprotocol Extensions)
- RFC 2918 (Route Refresh Capability)
- RFC 3392 (BGP4 Capability Advertisement)
- Draft-ietf-idr-restart-10.txt (restart mechanism for BGP4)

Memory considerations

BGP4 can handle a very large number of routes and therefore requires a lot of memory. For example, in a typical configuration with a single BGP4 neighbor, a BGP4 device may need to hold up to 150,000 routes. Many configurations, especially those involving more than one neighbor, can require the device to hold even more routes. The device provides dynamic memory allocation for BGP4 data. BGP4 devices automatically allocate memory when needed to support BGP4 neighbors, routes, and route attribute entries. Dynamic memory allocation is performed automatically by the software and does not require a reload.

As a guideline, a device with a 2 GB Management 4 module can accommodate 150 – 200 neighbors, with the assumption that the device receives about one million routes total from all neighbors, and sends about eight million routes total to neighbors. For each additional one million incoming routes, the capacity for outgoing routes decreases by about two million.

Grouping of RIB-out peers

To improve efficiency in the calculation of outbound route filters, the device groups BGP4 peers together based on their outbound policies. To reduce RIB-out memory usage, the device then groups the peers within an outbound policy group according to their RIB-out routes. All peers sharing a single RIB-out route (up to 32 peers per group) also share a single physical RIB-out entry, resulting in as much as a 30-fold memory usage reduction.

NOTE
RIB-out peer grouping is not shared between different VRFs or address families, and is not supported for VPNV4 or L2VPN peers.

BGP4 Restart

The Restart feature supports high-availability routing. With this feature enabled, disruptions in forwarding are minimized and route flapping is diminished to provide continuous service during the time that a device is performing a restart.

Under normal operation, restarting a BGP4 device causes the network to be reconfigured. In this situation, routes available through the restarting device are first deleted when the device goes down, and are then rediscovered and added back to the routing tables when the device is back up and running. In a network with devices that regularly restart, performance can degrade significantly and limit availability of network resources. BGP4 restart dampens the network response and limits route flapping by allowing routes to remain available between devices during a restart. BGP4 Restart operates between a device and peers, and must be configured on each participating device.

A BGP4 device with Restart enabled advertises this capability to establish peering relationships with other devices. After a restart begins, all of the routes from the restarting device are marked as stale by neighbor devices, but continue to be used for the length of time configured for the restart timer. After the device is restarted, it begins to receive routing updates from the peers. When the restarting device receives the end-of-RIB marker that indicates it has received all of the BGP4 route updates, all of the routes are recomputed and newly computed routes replace the routes labeled as stale in the route map. If the device does not come back up within the time configured for the purge timer, the routes marked stale are removed.

NOTE
A second management module must be installed for the device to function as a restart device. If the device functions as a restart helper device only, there is no requirement for a secondary management module.

The implementation of BGP4 Restart supports the following Internet Draft:

- Draft-ietf-idr-restart-10.txt: restart mechanism for BGP4

For details concerning configuration of the BGP4 Restart feature, refer to "Configuring BGP4 Restart" on page 1098.

BGP4 Peer notification during a management module switchover

The BGP4 Peer notification process restores BGP4 adjacency quickly and allows packet forwarding between the newly active management module and the BGP4 peers. The handling of TCP packets with an MD5 digest prevents the silent dropping of TCP packets without triggering a RESET packet.

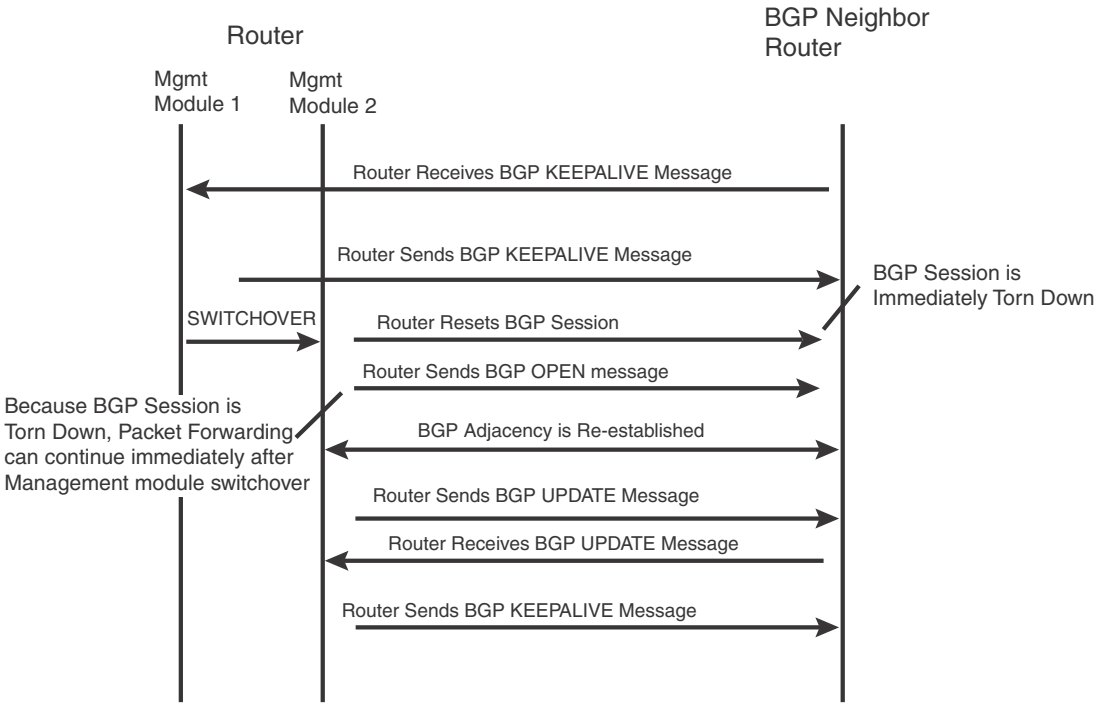
The BGP4 peer notification process operates effectively when implemented for the following processes that involve the intentional switching of the active status from one management module to another:

- **System Reload** – When a device undergoes the reload process, both management modules and all interface modules are rebooted. All BGP4 sessions are terminated BEFORE the system triggers the hardware reset.
- **Switchover Requested by User** – Switching over to a standby management module can be triggered by the **switchover**, **reset**, **reload**, and **hitless-reload** commands. When these commands are executed, the active management module resets the BGP4/TCP sessions with BGP4 neighbors before transferring control to the standby management module.

NOTE
Graceful-restart-enabled BGP4 sessions are not reset. The BGP4 graceful-restart protocol allows a BGP4 session to reconnect gracefully without going through the normal process.

Figure 147 describes the procedure used between the management modules in a device and a BGP4 neighbor device.

FIGURE 147 Management module switchover behavior for BGP4 peer notification



If the active management module fails due to a fault, the management module does not have the opportunity to reset BGP4 sessions with neighbors as described for intentional failovers, and illustrated in Figure 147. In this situation the management module will reboot, or the standby management module becomes the new active management module. Since the new active management module does not have the TCP/BGP4 information needed to reset the previous sessions, a remote BGP4 peer session is only reset when it sends a BGP4/TCP keep-alive packet to this device, or when the BGP4 hold-time expires.

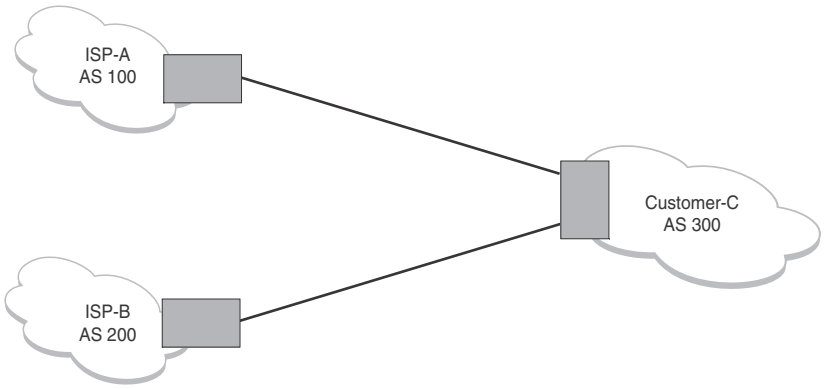
To help reduce the reconnection time after a management module failover or system reload, if an incoming TCP packet contains an MD5 digest, and no matching TCP session is found, the device attempts to find a matching BGP4 peer based on the IP address. If a BGP4 peer configuration can be found, the device looks up the MD5 password configured for the peer, and uses it to send a RESET packet.

BGP4 neighbor local AS

This feature allows you to configure a device so that it adds a peer to an AS that is different from the AS to which it actually belongs. This feature is useful when an ISP is acquired by another ISP. In this situation, customers of the acquired ISP might not want to (or might not be able to) adjust their configuration to connect to the AS of the acquiring provider.

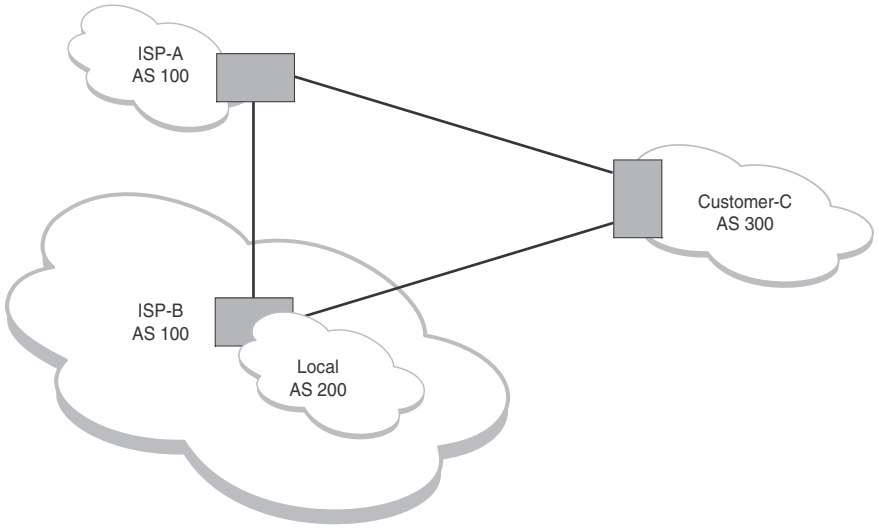
For example in Figure 168, Customer C is connected to ISP-A which is in AS 100 and ISP-B which is in AS 200.

FIGURE 148 Example of customer connected to two ISPs



In the example shown in Figure 149, ISP-A has purchased ISP-B. The AS associated with ISP-B changes to AS 100. If Customer C cannot or does not want to change their configuration or peering relationship with ISP-B, a peer with Local-AS configured with the value 200 can be established on ISP-B.

FIGURE 149 Example of Local AS configured on ISP-B



A Local AS is configured using the BGP4 **neighbor** command, as described in "Configuring BGP4 neighbors" on page 1037. To confirm that a Local AS has been configured use the **show ip BGP4 neighbor** command, as described in "Displaying BGP4 neighbor information" on page 1109.

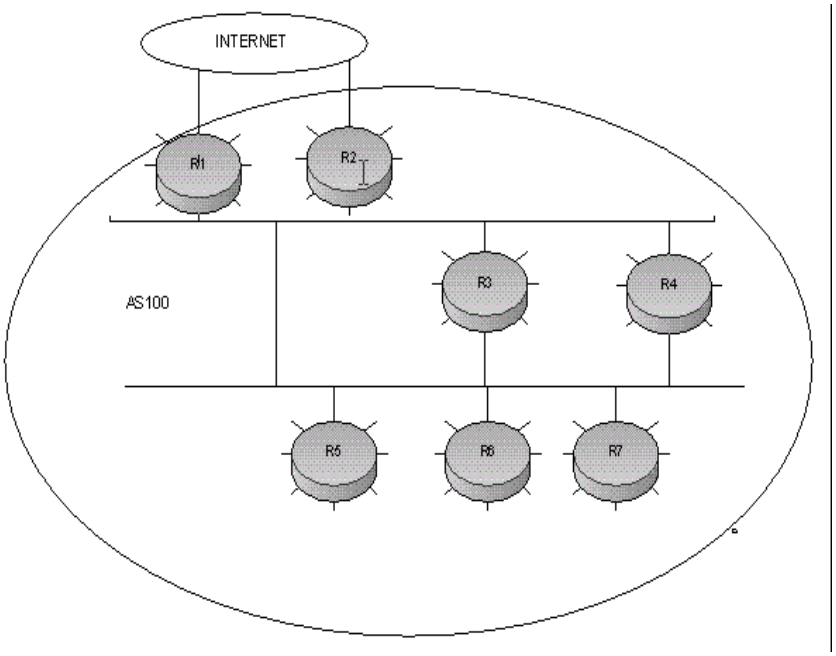
BGP4 null0 routing

BGP4 considers the null0 route in the routing table (for example, static route) as a valid route, and can use the null0 route to resolve the next hop. If the next hop for BGP4 resolves into a null0 route, the BGP4 route is also installed as a null0 route in the routing table.

The null0 routing feature allows network administrators to block certain network prefixes using null0 routes and route-maps, directing a remote device to drop all traffic for a network prefix by redistributing a null0 route into BGP4.

Figure 150 shows a topology for a null0 routing application example.

FIGURE 150 SAMPLE null0 routing application



Refer to "Configuring BGP4 null0 routing" on page 1099 for an example of how to configure a null0 routing application to stop denial of service attacks from remote hosts on the Internet

Configuring BGP4

Once you activate BGP4, you can configure the BGP4 options. There are two configuration levels: global and address family.

At the *global level*, all BGP4 configurations apply to IPv4 and IPv6. Enter this layer using the **device BGP4** command

Under the global level, you specify an ***address family***. Address families separate IPv4 and IPv6 BGP4 configurations. Go to this level by entering the **address-family** command at the device BGP4 level. The command requires you to specify the IPv4 or IPv6 network protocol.

The **address family** command also requires you to select a sub-address family, which is the type of routes for the configuration. Specify multicast or unicast routes.

FIGURE 151 BGP4 configuration levels

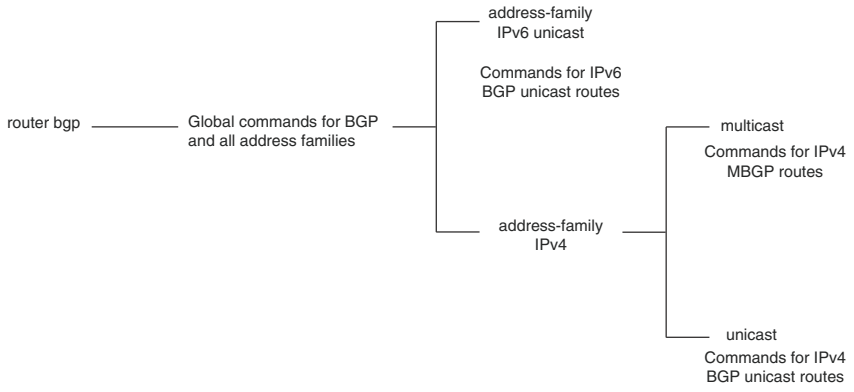


Table • shows the commands that are available at the various BGP4 configuration levels.

TABLE 166 IPv4 BGP4 commands for different configuration levels

Command	Global (IPv4 and IPv6)	IPv4 address family unicast	IPv4 address family multicast	See
address-family	x	x	x	"Entering and exiting the address family configuration level" on page 1013
aggregate-address		x	x	"Aggregating routes advertised to BGP4 neighbors" on page 1013
always-compare-med	x			"Configuring the device to always compare Multi-Exit Discriminators" on page 1014
as-path-ignore	x			"Disabling or re-enabling comparison of the AS-Path length" on page 1015
bgp-redistribute-internal		x		"Redistributing IBGP routes" on page 1015
client-to-client-reflection				"Disabling or re-enabling client-to-client route reflection" on page 1016
cluster-id	x			"Configuring a route reflector" on page 1016
compare-med-empty-as path	x			"Configuring the device to always compare Multi-Exit Discriminators" on page 1014
compare-routerid	x			"Enabling or disabling comparison of device IDs" on page 1016
confederation	x			"Configuring confederations" on page 1017
dampening		x	x	"Configuring route flap dampening" on page 1028
default-information-originate		x	x	"Originating the default route" on page 1029
default-local-preference	x			"Changing the default local preference" on page 1029
default-metric		x	x	"Changing the default metric used for redistribution" on page 1030

TABLE 166 IPv4 BGP4 commands for different configuration levels (Continued)

Command	Global (IPv4 and IPv6)	IPv4 address family unicast	IPv4 address family multicast	See
distance	x			"Changing administrative distances" on page 1030
enforce-first-as	x			"Requiring the first AS to be the neighbor AS" on page 1032
exit-address-family	x	x	x	"Entering and exiting the address family configuration level" on page 1013
fast-external-fallover	x			"Enabling fast external fallover" on page 1033
local-as	x			"Setting the local AS number" on page 1033
maximum-paths		x		"Configuring BGP4 multipath load sharing" on page 1034
med-missing-as-worst	x			"Configuring paths without MEDs as the least favorable" on page 1037
multipath		x		"Configuring paths without MEDs as the least favorable" on page 1037
neighbor	x	x	x	"Configuring BGP4 neighbors" on page 1037 "Configuring a BGP4 peer group" on page 1046
network		x	x	"Specifying a list of networks to advertise" on page 1049
next-hop-enable-default		x		"Using the IP default route as a valid next-hop for a BGP4 route" on page 1051
next-hop-recursion		x		"Enabling next-hop recursion" on page 1051
redistribute		x	x	"Modifying redistribution parameters" on page 1054
show	x	x	x	"Displaying BGP4 information" on page 1104
table-map		x	x	"Using a table map to set the tag value" on page 1057
timers	x			"Changing the Keep Alive Time and Hold Time" on page 1057
update-time		x	x	"Changing the BGP4 next-hop update timer" on page 1058

Parameter changes that take effect immediately

The following parameter changes take effect immediately:

- Enable or disable BGP4.
- Set or change the local AS.
- Add neighbors.
- Change the update timer for route changes.
- Disable or enable fast external fallover.

- Specify individual networks that can be advertised.
- Change the default local preference, default information originate setting, or administrative distance.
- Enable or disable use of a default route to resolve a BGP4 next-hop route.
- Enable or disable MED (metric) comparison.
- Require the first AS in an update from an EBGP neighbor to be the neighbor AS.
- Change MED comparison parameters.
- Disable comparison of the AS-Path length.
- Enable comparison of the device ID.
- Enable next-hop recursion.
- Change the default metric.
- Disable or re-enable route reflection.
- Configure confederation parameters.
- Disable or re-enable load sharing.
- Change the maximum number of load sharing paths.
- Change other load-sharing parameters.
- Define route flap dampening parameters.
- Add, change, or negate redistribution parameters (except changing the default MED; see below).
- Add, change, or negate route maps (when used by the **network** command or a redistribution command).
- Aggregate routes.
- Apply maximum AS path limit settings for UPDATE messages.

Parameter changes that take effect after resetting neighbor sessions

The following parameter changes take effect only after the BGP4 sessions on the device are cleared, or reset using the "soft" clear option. (Refer to "Closing or resetting a neighbor session" on page 1095.)

- Change the Hold Time or Keep Alive Time.
- Add, change, or negate filter tables that affect inbound and outbound route policies.
- Apply maximum AS path limit settings to the RIB.

Parameter changes that take effect after disabling and re-enabling redistribution

Changing the default MED (metric) change takes effect only after you disable and then re-enable redistribution:

Enabling and disabling BGP4

BGP4 is disabled by default. To enable BGP4, you must perform the following steps.

1. Enable the BGP4 protocol.
2. Set the local AS number.

NOTE
BGP4 is not functional until you specify the local AS number.

3. Add each BGP4 neighbor (peer BGP4 device) and identify the AS the neighbor is in.
4. Save the BGP4 configuration information to the system configuration file.

For example, enter commands such as the following.

```
NetIron> enable
NetIron# configure terminal
NetIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
NetIron(config-bgp)# local-as 10
NetIron(config-bgp)# write memory
```

Syntax: router bgp

The **router bgp** command enables the BGP4 protocol.

For information on the local AS number, refer to "Setting the local AS number" on page 1033.

NOTE
By default, the Dell device ID is the IP address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default device ID is the lowest numbered IP interface address configured on the device. For more information, refer to "Changing the device ID" on page 1058. If you change the device ID, all current BGP4 sessions, OSPF adjacencies, and OSPFv3 adjacencies are cleared.

NOTE
When BGP4 is enabled on a device, resetting the system is unnecessary. The protocol is activated when you enable it. The device begins a BGP4 session with a BGP4 neighbor when you add the neighbor.

Disabling BGP4

If you disable BGP4, the device removes all the running configuration information for the disabled protocol from the running configuration. To restore the BGP4 configuration, you must reload the software to load the BGP4 configuration from the startup configuration. When you save the startup configuration file after disabling the protocol, all of the BGP4 configuration information for the disabled protocol is removed from the startup configuration file.

The CLI displays a warning message such as the following.

```
NetIron(config)# no router bgp
router bgp mode now disabled. All bgp config data will be lost when writing to
flash!
```

The Web Management Interface does not display a warning message.

If you are testing a BGP4 configuration and may need to disable and re-enable the protocol, you should make a backup copy of the startup configuration file containing the BGP4 configuration information. If you remove the configuration information by saving the configuration after disabling the protocol, you can restore the BGP4 configuration by copying the backup copy of the startup configuration file onto the flash memory.

To disable BGP4 without losing the BGP4 configuration information, remove the local AS (for example, by entering the **no local-as** <*num*> command). When you remove the local AS, BGP4 retains the other configuration information but will not become operational until you reset the local AS.

Entering and exiting the address family configuration level

The BGP4 address family contains a unicast or multicast sub-level.

To go to the IPv4 BGP4 unicast address family configuration level, enter the following command.

```
NetIron(config-bgp)# address-family ipv4 unicast
NetIron(config-bgp)#
```

NOTE
The CLI prompt for the global BGP4 level and the BGP4 address-family IPv4 unicast level is the same.

To go to the IPv4 BGP4 multicast address family configuration level, enter the following command.

```
NetIron(config-bgp)# address-family ipv4 multicast
NetIron(config-bgp-ipv4m)#
```

Syntax: [**no**] **address-family ipv4 unicast** [**vrf** <*vrf-name*>] | **ipv4 multicast**

The default is the ipv4 unicast address family level.

The **vrf** option allows you to configure a unicast instance for the VRF specified by the <*vrf-name*> variable.

To exit an address family configuration level, enter the following command.

```
NetIron(config-bgp-ipv6u)# exit-address-family
NetIron(config-bgp)#
```

Syntax: [**no**] **exit-address-family**

Aggregating routes advertised to BGP4 neighbors

By default, the device advertises individual routes for all networks. The aggregation feature allows you to configure the device to aggregate routes from a range of networks into a single network prefix. For example, without aggregation, the device will individually advertise routes for networks 207.95.1.0/24, 207.95.2.0/24, and 207.95.3.0/24. You can configure the device to end a single, aggregate route for the networks instead. The aggregate route can be advertised as 207.95.0.0/16.

To aggregate routes for 209.157.22.0/24, 209.157.23.0/24, and 209.157.24.0/24, enter the following command.

```
NetIron(config-bgp)# aggregate-address 209.157.0.0 255.255.0.0
```

Syntax: [**no**] **aggregate-address** <*ip-addr*> <*ip-mask*> [**as-set**] [**summary-only**] [**suppress-map** <*map-name*>] [**advertise-map** <*map-name*>] [**attribute-map** <*map-name*>]

The <*ip-addr*> and <*ip-mask*> parameters specify the aggregate value for the networks. Specify 0 for the host portion and for the network portion that differs among the networks in the aggregate. For example, to aggregate 10.0.1.0/24, 10.0.2.0/24, and 10.0.3.0/24, enter the IP address 10.0.0.0 and the network mask 255.255.0.0.

The **as-set** parameter causes the device to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.

The **summary-only** parameter prevents the device from advertising more specific routes contained within the aggregate route.

The **suppress-map** <*map-name*> parameter prevents the more specific routes contained in the specified route map from being advertised.

The **advertise-map** <*map-name*> parameter configures the device to advertise the more specific routes in the specified route map.

The **attribute-map** <*map-name*> parameter configures the device to set attributes for the aggregate routes based on the specified route map.

NOTE
For the **suppress-map**, **advertise-map**, and **attribute-map** parameters, the route map must already be defined. Refer to "“Defining route maps” on page 1068 for information on defining a route map.

Configuring the device to always compare Multi-Exit Discriminators

A Multi-Exit Discriminator (MED) is a value that the BGP4 algorithm uses when it compares multiple paths received from different BGP4 neighbors in the same AS for the same route. In BGP4, a MED for a route is equivalent to its metric.

BGP4 compares the MEDs of two otherwise equivalent paths ***if and only if*** the routes were learned from the same neighboring AS. This behavior is called ***deterministic MED***. Deterministic MED is always enabled and cannot be disabled.

You can enable the device to always compare the MEDs, regardless of the AS information in the paths. For example, if the device receives UPDATES for the same route from neighbors in three ASs, the device can compare the MEDs of all the paths together instead of comparing the MEDs for the paths in each AS individually.

To enable this comparison, enter the **always-compare-med** command at the BGP4 configuration level of the CLI. This option is disabled by default.

By default, value 0 (most favorable) is used in MED comparison when the MED attribute is not present. The default MED comparison results in the device favoring route paths that do not have their MEDs. Use the **med-missing-as-worst** command to force the device to regard a BGP4 route with a missing MED attribute as the least favorable route.

NOTE
MED comparison is not performed for internal routes originated within the local AS or confederation unless the **compare-med-empty-aspath** command is configured.

To configure the device to always compare MEDs, enter the following command.

```
NetIron(config-bgp)# always-compare-med
```

Syntax: [no] **always-compare-med**

The following BGP4 command directs BGP4 to take the MED value into consideration even if the route has an empty as-path path attribute.

```
NetIron(config) router bgp
NetIron(config-bgp-router)# compare-med-empty-aspath
```

Syntax: [no] **compare-med-empty-aspath**

Disabling or re-enabling comparison of the AS-Path length

AS-Path comparison is Step 5 in the algorithm that BGP4 uses to select the next path for a route. Comparison of the AS-Path length is enabled by default. To disable it, enter the following command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# as-path-ignore
```

Syntax: [no] **as-path-ignore**

This command disables comparison of the AS-Path lengths of otherwise equal paths. When you disable AS-Path length comparison, the BGP4 algorithm shown in "How BGP4 selects a path for a route" on page 1000 skips from Step 4 to Step 6.

Redistributing IBGP routes

By default, the device does not redistribute IBGP routes from BGP4 into RIP, OSPF, or ISIS. This behavior helps eliminate routing loops. However, if your network can benefit from redistributing IBGP routes from BGP4 into OSPF, ISIS or RIP, you can enable the device to redistribute the routes.

To enable the device to redistribute BGP4 routes into OSPF, RIP, or ISIS, enter the following command.

```
NetIron(config-bgp)# bgp-redistribute-internal
```

Syntax: [no] **bgp-redistribute-internal**

To disable redistribution of IBGP routes into RIP, ISIS, and OSPF, enter the following command.

```
NetIron(config-bgp)# no bgp-redistribute-internal
```

Disabling or re-enabling client-to-client route reflection

By default, the clients of a route reflector are not required to be fully meshed. Routes from a client are reflected to other clients. However, if the clients are fully meshed, route reflection is not required between clients.

If you need to disable route reflection between clients, enter the **no client-to-client-reflection** command. When this feature is disabled, route reflection does not occur between clients does still occur between clients and non-clients.

```
NetIron(config-bgp)# no client-to-client-reflection
```

Enter the following command to re-enable the feature.

```
NetIron(config-bgp)# client-to-client-reflection
```

Syntax: [**no**] **client-to-client-reflection**

Configuring a route reflector

You can configure one cluster ID on the device so that all route-reflector clients for the device become members of the cluster.

To configure a device with cluster id 1, enter the following command.

```
NetIron(config-bgp)# cluster-id 1
```

Syntax: [**no**] **cluster-id** <*num*> | <*ip-addr*>

The <*num*> | <*ip-addr*> parameters specify the cluster ID (1 – 4294967295) or an IP address. The default is the device ID.

NOTE
If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops within the cluster.

Enabling or disabling comparison of device IDs

Device ID comparison is Step 11 in the algorithm BGP4 uses to select the next path for a route.

NOTE
Comparison of device IDs is applicable only when BGP4 load sharing is disabled.

When device ID comparison is enabled, the path comparison algorithm compares the device IDs of the neighbors that sent the otherwise equal paths:

- If BGP4 load sharing is disabled (maximum-paths 1), the device selects the path that came from the neighbor with the lower device ID.

- If BGP4 load sharing is enabled, the device load shares among the remaining paths. In this case, the device ID is not used to select a path.

NOTE
Device ID comparison is disabled by default.

To enable device ID comparison, enter the **compare-routerid** command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# compare-routerid
```

Syntax: [**no**] **compare-routerid**

For more information, refer to "How BGP4 selects a path for a route" on page 1000.

Configuring confederations

A **confederation** is a BGP4 Autonomous System (AS) that has been subdivided into multiple, smaller ASs. Subdividing an AS into smaller ASs simplifies administration and reduces BGP4-related traffic, which in turn reduces the complexity of the Interior Border Gateway Protocol (IBGP) mesh among the BGP4 devices in the AS.

The implementation of this feature is based on RFC 3065.

Normally, all BGP4 devices within an AS must be fully meshed, so that each BGP4 device has BGP4 sessions to all the other BGP4 devices within the AS. This is feasible in smaller ASs, but becomes unmanageable in ASs containing many BGP4 devices.

When you configure BGP4 devices into a confederation, all the devices within a sub-AS (a subdivision of the AS) use IBGP and must be fully meshed. However, devices use EBGP to communicate between different sub-ASs.

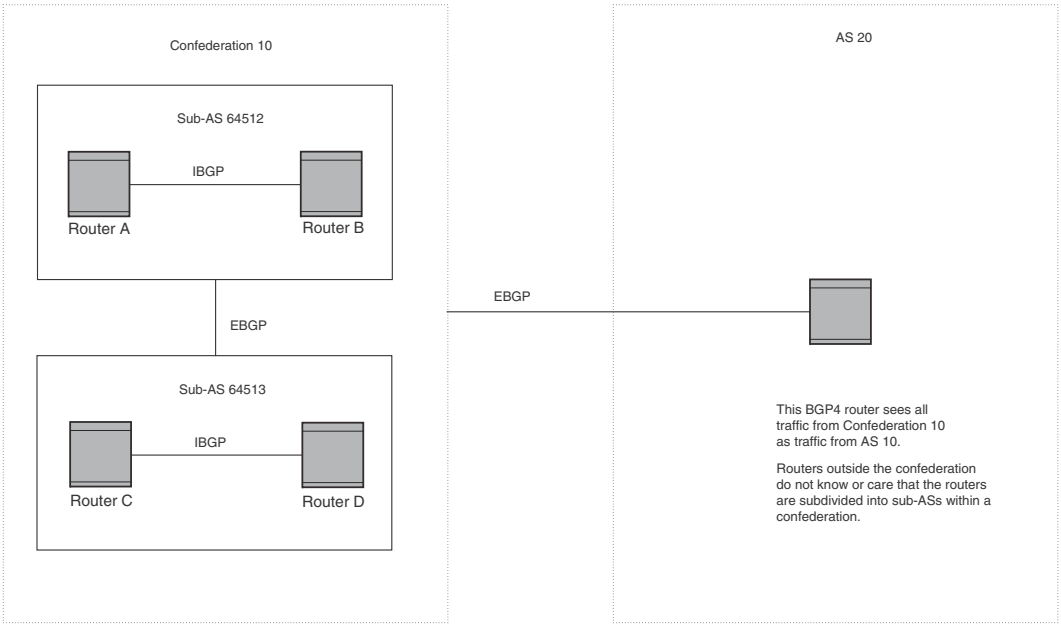
Another way to reduce the complexity of an IBGP mesh is to use route reflection. However, if you want to run different Interior Gateway Protocols (IGPs) within an AS, you must configure a confederation. You can run a separate IGP within each sub-AS.

To configure a confederation, configure groups of BGP4 devices into sub-ASs. A sub-AS is simply an AS. The term "sub-AS" distinguishes ASs within a confederation from ASs that are not in a confederation. For the viewpoint of remote ASs, the confederation ID is the AS ID. Remote ASs do not know that the AS represents multiple sub-ASs with unique AS IDs.

NOTE
You can use any valid AS numbers for the sub-ASs. If your AS is connected to the Internet, it is recommended that you use numbers from within the private AS range (64512 – 65535). These are private ASs numbers and BGP4 devices do not propagate these AS numbers to the Internet.

Figure 152 shows an example of a BGP4 confederation.

FIGURE 152 Example BGP4 confederation



In this example, four devices are configured into two sub-ASs, each containing two of the devices. The sub-ASs are members of confederation 10. Devices within a sub-AS must be fully meshed and communicate using IBGP. In this example, devices A and B use IBGP to communicate. Devices C and D also use IBGP. However, the sub-ASs communicate with one another using EBGP. For example, device A communicates with device C using EBGP. The devices in the confederation communicate with other ASs using EBGP.

Devices in other ASs are unaware that devices A – D are configured in a confederation. In fact, when devices in confederation 10 send traffic to devices in other ASs, the confederation ID is the same as the AS number for the devices in the confederation. Thus, devices in other ASs see traffic as coming from AS 10 and are unaware that the devices in AS 10 are subdivided into sub-ASs within a confederation.

Configuring a BGP4 confederation

To configure a BGP4 configuration, perform these configuration tasks on each BGP4 device within the confederation:

- Configure the local AS number. The local AS number indicates membership in a sub-AS. All BGP4 devices with the same local AS number are members of the same sub-AS. BGP4 devices use the local AS number when communicating with other BGP4 devices in the confederation.
- Configure the confederation ID. The confederation ID is the AS number by which BGP4 devices outside the confederation recognize the confederation. A BGP4 device outside the confederation is not aware of, and does not care that BGP4 devices are in multiple sub-ASs. A BGP4 device uses the confederation ID to communicate with devices outside the confederation. The confederation ID must differ from the sub-AS numbers.
- Configure the list of the sub-AS numbers that are members of the confederation. All devices within the same sub-AS use IBGP to exchange device information. Devices in different sub-ASs within the confederation use EBGP to exchange device information.

The following command examples show how to implement the confederation shown in Figure 152.

To configure four device devices to be members of confederation 10 (consisting of sub-ASs 64512 and 64513), enter commands such as the following.

Commands for Device A

```
NetIronA(config)# router bgp
NetIronA(config-bgp)# local-as 64512
NetIronA(config-bgp)# confederation identifier 10
NetIronA(config-bgp)# confederation peers 64512 64513
NetIronA(config-bgp)# write memory
```

Syntax: [**no**] **local-as** <*num*>

The <*num*> parameter with the **local-as** command indicates the AS number for the BGP4 devices within the sub-AS. You can specify a number in the range 1 – 4294967295. It is recommended that you use a number within the range of well-known private ASs, 64512 – 65535.

Syntax: [**no**] **confederation identifier** <*num*>

The <num> parameter with the **confederation identifier** command indicates the confederation number. The confederation ID is the AS number by which BGP4 devices outside the confederation recognize the confederation. A BGP4 device outside the confederation is not aware of, and does not care that your BGP4 devices are in multiple sub-ASs. BGP4 devices use the confederation ID when communicating with devices outside the confederation. The confederation ID must be different from the sub-AS numbers. For the <*num*> parameter, you can specify a number in the range 1 – 4294967295.

Syntax: [**no**] **confederation peers** <*num*> [<*num*> ...]

The <*num*> parameter with the **confederation peers** command indicates the sub-AS numbers for the sub-ASs in the confederation. You can list all sub-ASs in the confederation. You must specify all the sub-ASs with which this device has peer sessions in the confederation. All the devices within the same sub-AS use IBGP to exchange device information. Devices in different sub-ASs within the confederation use EBGP to exchange device information. The <*num*> is a number in the range 1 – 4294967295.

Commands for Device B

```
NetIronB(config)# router bgp
NetIronB(config-bgp)# local-as 64512
NetIronB(config-bgp)# confederation identifier 10
NetIronB(config-bgp)# confederation peers 64512 64513
NetIronB(config-bgp)# write memory
```

Commands for Device C

```
NetIronC(config)# router bgp
NetIronC(config-bgp)# local-as 64513
NetIronC(config-bgp)# confederation identifier 10
NetIronC(config-bgp)# confederation peers 64512 64513
NetIronC(config-bgp)# write memory
```

Commands for Device D

```
NetIronD(config)# router bgp
NetIronD(config-bgp)# local-as 64513
NetIronD(config-bgp)# confederation identifier 10
NetIronD(config-bgp)# confederation peers 64512 64513
NetIronD(config-bgp)# write memory
```

Four-byte Autonomous System Numbers (AS4)

This section describes the reasons for enabling four-byte autonomous system numbers (AS4s). AS4s are supported by default. You can specify and view AS4s by default and using the enable facility described in this section. However, because not all devices in a network are always capable of utilizing AS4s, the act of enabling them on the local device initiates a facility for announcing the capability and negotiating its use with neighbors. If you do not enable AS4s on a device, other devices do not know that this device is sending them.

The system uses a hierarchy to prioritize the utilization of the AS4 capability. The prioritization depends on the CLI configuration commands. AS4s can be enabled and configured at the level of a neighbor, a peer group, or globally for the entire device, according to the following bottom-up hierarchy:

- If a neighbor has no configuration for AS4s but it belongs to a peer group, the neighbor uses the configuration from the peer group. For example, if you configure a neighbor but do not include a specification for AS4s, one of the following applies:
  - The neighbor uses the AS4 configuration for a peer group if it belongs to a peer group.
  - The neighbor uses the device configuration if it does not belong to a peer group or the peer group has no AS4 configuration.
- If a peer group has no configuration for AS4s, it can use the global configuration of the device. If the device has no configuration for AS4s, then a neighbor or peer group without a configuration for AS4s use the device default—no announcement or negotiation of AS4s.
- If a neighbor belongs to peer group with an AS4 configuration but you want that neighbor to be disabled or have a different AS4 configuration, the neighbor AS4 configuration overrides the peer group configuration. For example, you can ensure that neighbor has no AS4 announcement and negotiation activity even though the peer group is enabled for AS4 capability.

NOTE
The configuration for AS4 can be enabled, disabled, or can have no explicit configuration.

CLI commands allow you to disable AS4s on an entity whose larger context has AS4s enabled. For example, you can use a CLI command to disable AS4s on a neighbor that is a member of a peer group that is enabled for AS4s. Refer to "Enabling AS4 numbers" on page 1021.

Normally, AS4s are sent only to a device, peer group, or neighbor that is similarly configured for AS4s. If a AS4 is configured for a local-AS, the system signals this configuration by sending AS_TRANS in the My Autonomous System field of the OPEN message. However, if the AS4 capability for a neighbor is disabled, the local device does not send the four-octet Autonomous System number capability to the neighbor.

Enabling AS4 numbers

This section describes how to enable the announcement and negotiation of AS4s and describes the different types of notation that you can use to represent a AS4.

You can enable AS4s on a device, a peer group, and a neighbor. For global configuration, the **capability** command in the BGP4 configuration context enables or disables AS4 support. For a peer group or a neighbor, **capability** is a keyword for the **neighbor** command. In addition to enabling AS4s for a neighbor or a peer group, you can also use the combination of the **capability** keyword and the optional **enable** or **disable** keyword to disable this feature in a specific case where the AS4s are enabled for a larger context. The "Neighbor configuration of AS4s"section illustrates this capability.

Global AS4 configuration

To enable AS4s globally, use the **capability** command in the BGP4 configuration context, as shown.

```
NetIron(config-bgp)# capability as4 enable
```

Syntax: [**no**] **capability as4 enable** | **disable**

The no form of the **capability** command deletes the announcement and negotiation configuration of AS4s (if it has been enabled) at the global level. Using the regular form of the command with the **disable** keyword has the same effect on the global configuration. Disabling or using the no form of the command does not affect the configuration at the level of a peer or neighbor.

The consequences of choosing between the **enable** or **disable** keyword are reflected in the output of the **show running configuration** command.

Peer group configuration of AS4s

To enable AS4s for a peer group, use the **capability** keyword with the **neighbor** command in the BGP4 configuration context, as the following example for the Peergroup_1 peer group illustrates.

```
NetIron(config-bgp)# neighbor Peergroup_1 capability as4 enable
```

Syntax: [**no**] **neighbor** <*peer-group-name*> **capability as4 enable** | **disable**

The **no** form of the **neighbor** command along with the **capability as4** keywords disables the announcement and negotiation of AS4s in the named peer group. Using the regular form of the command with the **disable** keyword has the same effect on the neighbor configuration.

The consequences using the **enable** or **disable** keywords are reflected in the output of the **show running configuration** command. However, if the peer group configuration omits an explicit AS4 argument, the **show running configuration** output will not contain AS4 information.

Neighbor configuration of AS4s

To enable AS4s for a neighbor, use the **capability** and **as4** keywords with the **neighbor** command in the BGP4 configuration context, as the following example for IP address 1.1.1.1 illustrates.

```
NetIron(config-bgp)# neighbor 1.1.1.1 capability as4 enable
```

Syntax: [**no**] **neighbor** <*IP address*> **capability as4 enable** | **disable**

The **no** form of the **neighbor** command with the **capability as4** keywords deletes the neighbor-enable for AS4s.

The consequences of using the **enable** or **disable** keywords are reflected in the output of the **show running configuration** command. However, if the neighbor configuration omits an explicit AS4 argument, the **show running configuration** output will not contain AS4 information.

To disable AS4s on a particular neighbor within a peer group that is enabled for AS4s, enter a command similar to the following.

```
NetIron(config-bgp)# neighbor 1.1.1.1 capability as4 disable
```

Specifying the local AS number

The local autonomous system number (ASN) identifies the AS where the BGP4 device resides.

Normally, AS4s are sent only to a device, peer group, or neighbor that is similarly configured for AS4s. Typically, if you try to set up a connection from an AS4-enabled device to a device that processes only two-byte ASNs, the connection fails to come up unless you specify the reserved ASN 23456 as the local ASN to send to the far-end device.

To set the local AS number, enter commands such as the following.

```
NetIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
NetIron(config-bgp)# local-as 100000
NetIron(config-bgp)# write memory
```

Syntax: [**no**] **local-as** <*num*>

The <*num*> parameter specifies a local ASN in the range 1 – 4294967295. No default exists for <*num*>. ASNs 64512 – 65535 are the well-known private BGP4 AS numbers and are not advertised to the Internet community.

Route-map set commands and AS4s

You can prepend a AS4 number to an AS path or make the AS number a tag attribute for a route map, as shown here.

```
NetIron(config-routemap test)# set as-path prepend 7701000
```

Syntax: [**no**] **set as-path prepend** <*num, num*, ...> | **tag**

Use the **no** form of this command to remove the configuration.

NOTE
If the AS path for a route map has prepended ASNs and you want to use the **no** form of the command to delete the configuration, you must include the prepended ASNs in the **no set as-path** entry. For example, if 70000 and 70001 have been prepended to a route map, enter **no set as-path prepend 70000 70001**. As a shortcut, in the configuration context of a particular route map, you can also copy and paste ASNs from the output of **show** commands, such as show **route-map** or **show ip bgp route**.

Use the **prepend** keyword to prepend one or more ASNs. The maximum number of ASNs that you can prepend is 16. The range for each ASN is 1 – 4294967295.

Entering the **tag** keyword sets the tag as an AS-path attribute.

You can specify a route target (art) or a site of origin (soo) for an extended community, as shown in the following example.

```
NetIron(config-routemap test)# set extcommunity rt 7701000:10
```

Syntax: [**no**] **set extcommunity rt** <*asn:nn* | *ip-address:nn*> | **soo** <*asn:nn* | *ip-address:nn*>

The **rt** keyword specifies a route target in the form of a route ID. The route ID can be an ASN or IP address. The second part of the route ID is a user-specific numeric variable *nn*. The ASN can be a maximum of 4 bytes (in the range 1 – 4294967295). If you specify an AS4 or IP address, the *nn* variable is limited to a maximum length of 2 bytes. If the feature for announcing and negotiating AS4 is disabled, *nn* can be 4 bytes.

The **soo** keyword specifies a site or origin in the form of a route ID. The route ID can be an ASN4 or IP address. The second part of the route ID is a user-specific numeric variable *nn*. The AS4 can be a maximum of 4 bytes (in the range 1 – 4294967295). If you specify an AS4 or IP address, the *nn* variable is limited to a maximum length of 2 bytes. If the feature for announcing and negotiating AS4 is disabled, *nn* can be 4 bytes.

Clearing BGP4 routes to neighbors

You can clear BGP4 connections using the AS4 as an argument with the **clear ip bgp neighbor** command in the configuration context level of the CLI. as shown.

```
NetIron(config)# clear ip bgp neighbor 80000
```

Syntax: **clear ip bgp neighbor** < **all** | <*ip-addr*> | <*peer-group-name*> | <*as-num*> >
[**last-packet-with-error** | **notification-errors** | [**soft** [**in** | **out**] | **soft-outbound**]

The neighbor specification is either **all,** <*ip-addr*>, <*peer-group-name*>, or <*as-num*>. The **all** parameter specifies all neighbors. The <*ip-addr*> parameter specifies a neighbor by its IP interface with the device. The <*peer-group-name*> specifies all neighbors in a specific peer group. The <*as-num*> parameter specifies all neighbors within the specified AS. After choosing one mandatory parameter, you can choose an optional parameter.

The **soft** [**in** | **out**] parameter determines whether to refresh the routes received from the neighbor or the routes sent to the neighbor. If you do not specify **in** or **out**, the device performs a soft refresh in both options:

- **soft in** performs one of the following actions on inbound routes, according to other configuration settings:
  - If you enabled soft reconfiguration for the neighbor or peer group, **soft in** updates the routes by comparing the route policies against the route updates that the device has stored. Soft reconfiguration does not request additional updates from the neighbor or otherwise affect the session with the neighbor.
  - If you did not enable soft reconfiguration, **soft in** requests the entire BGP4 route table on the neighbor (Adj-RIB-Out), then applies the filters to add, change, or exclude routes.
  - If a neighbor does not support dynamic refresh, **soft in** resets the neighbor session.
- **soft out** updates all outbound routes and then sends the entire BGP4 route table for the device (Adj-RIB-Out) to the neighbor after the device changes or excludes the routes affected by the filters.
- The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor.

NOTE
Use **soft-outbound** only if the outbound policy is changed. The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters. However, the device sends to the neighbor only the existing routes that are affected by the new or changed filters.The **soft out** parameter updates all outbound routes and then sends the entire BGP4 route table on the device to the neighbor after the device changes or excludes the routes affected by the filters.

AS4 notation

A AS4 can appear in either a plain or a dot notation format in the output of **show** commands. To select one of these formats, specify the format before entering the **show** command. This section defines these formats and describes how to select a format. The section "Formats of AS4s in show command output" on page 1134 contains examples of output in the various formats. The following notations are currently supported:

- With the default **asplain**, the ASN is a decimal integer in the range 1 – 4294967295.
- With **asdot+**, all ASNs are two integer values joined by a period character in the following format:

  <high order 16-bit value in decimal>.<low order 16-bit value in decimal>

  Using asdot+ notation, an AS number of value 65526 is represented as the string "0.65526," and an AS number of value 65546 is represented as the string "1.10."
- With **asdot,** an ASN less than 65536 uses asplain notation (and represents AS number values equal to or greater than 65536 using asdot+ notation). Using asdot notation, ASN 65526 is represented as the string "65526," and ASN 65546 is represented as the string "1.10".

NOTE
You can enter AS numbers in any format. However, if you want the asdot or asdot+ format to appear in the output of a **show** command, you must specify these in the CLI.

NOTE
Remember that AS path matching that uses regular expression is based on the configured AS format.

The following command sequences show how to enable the different notations for AS4s and how these notations appear in the output display.

To see ASNs in asplain, use the **show ip bgp** command.

```
NetIron(config)# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP,? - incomplete
Network            Next Hop        Metric LocPrf Weight Path
*>  47.1.1.0/24        192.168.1.5     1      100    0      90000 100 200 65535
65536 65537 65538 65539 75000 ?
```

To specify **asdot** notation before displaying IP BGP4 information, use the **as-format** command.

```
NetIron(config)# as-format asdot
NetIron(config)# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network            Next Hop        Metric LocPrf Weight Path
*>  47.1.1.0/24        192.168.1.5     1      100    0      1.24464 100 200 65535
1.0 1.1 1.2 1.3 1.9464 ?
```

Syntax: [no] **as-format asplain** | **asdot** | **asdot+**

The default is **asplain** and can be restored using the **no** version of the command, if the CLI is currently using **asdot** or **asdot+**.

To activate **asdot+** notation, enter **as-format asdot+** in the CLI.

```
NetIron(config)# as-format asdot+
NetIron(config)# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP,? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  47.1.1.0/24        192.168.1.5     1      100    0      1.24464 0.100 0.200
0.65535 1.0 1.1 1.2 1.3 1.9464 ?
```

BGP4 AS4 attribute errors

This section describes the handling of the confederation path segments in the AS4_PATH attribute, and also specifies the error handling for the new attributes.

To support AS4, the following attributes: AS4_PATH and AS4_Aggregator were specified in RFC 4893. Confederation path segments in an AS4_PATH are discarded and if there are any other errors such as: *attribute length, flag, confederation segments after AS_SEQ/AS_SET, Invalid segment types* and *More than one AS4_PATH* in these new attributes, the attribute is discarded and the error is logged.

Error logs

The device generates a log when it encounters attribute errors in AS4_PATH and AS4_AGGREGATOR.

NOTE
Logging of errors is rate-limited to not more than one message for every two minutes. Some errors may be lost due to this rate-limiting.

Sample log messages for various attribute errors are shown here.

Attribute length error (ignore the AS4_PATH)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received invalid
AS4_PATH attribute length (3) - entire AS4_PATH ignored
```

Attribute flag error (ignore the AS4_PATH)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received invalid
AS4_PATH attribute flag (0x40) - entire AS4_PATH ignored
```

Confederation segments after AS_SEQ/AS_SET (ignore the AS4_PATH)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received invalid
Confed info in AS4_PATH (@byte 43) - entire AS4_PATH ignored
```

Invalid segment types (ignore the AS4_ PATH)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received incorrect Seq
type/len in AS4_PATH (@byte 41) - entire AS4_PATH ignored
```

More than one AS4_PATH (Use the first one and ignore the others)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received multiple
AS4_PATH attributes - used first AS4_PATH attribute only
```

Specifying a maximum AS path length

You can use the **maxas-limit in** command to configure a router running BGP4 to discard routes that exceed a specified AS path limit. This limit can be configured globally, for peer groups, and for BGP neighbors.

When you configure the **maxas-limit in** setting, the behavior of the router changes to first check the length of the AS paths in the UPDATE messages and then to apply the inbound policy. If the AS path exceeds the configured length, then the router performs the following actions:

- Does not store the route in the RIB and does not forward the NLRIs and attributes contained in the UPDATE message for that route
- Logs an error
- Processes the withdrawn NLRIs in the same update message

If a route from a peer exceeds the configured Maximum AS path limit, the router also removes the same route from that peer, if it exists, from its own RIB.

After a maximum AS path length is configured, the maximum AS path limit applies to all new inbound routes. To update previously stored routes, you must perform an inbound soft reset for all of the address families activated for that particular BGP neighbor session.

NOTE
If the neighbor soft-reconfiguration feature is enabled, you must perform a hard reset on the router to impose the maximum length limit.

NOTE
Maxas-limit is checked against the received AS_PATH and AS4_PATH attributes.

BGP routers check for and, if configured, apply the **maxas-limit in** setting in the following order:

1. Neighbor value
2. Peer group value
3. Global value

In a case where a neighbor has no maximum AS limit, a peer group has a value of 3 configured, and the system has a value of 9 configured, all of the routers in the peer group will only use the peer group value; the global value will never be used.

Setting a global maximum AS path limit

The syntax for the global maximum AS path limit command is:

[no] maxas-limit in *<num>*

The **maxas-limit** keyword specifies the limit on the AS numbers in the as-path attribute. The **in** keyword allows the as-path attribute from any neighbor imposing a limit on AS numbers received. The default maximum length for the global system is 300. The range is 0 – 300. The **no** keyword removes the configuration at the global level.

NOTE
The router applies the BGP4 maximum AS path limit on a per virtual router basis.

To configure the global Maximum AS path limit to 15, enter the following command:

```
NetIron(config-bgp)# maxas-limit in 15
```

Setting a maximum AS path limit for a peer group or neighbor

To set maximum AS path limit for a peer group or a neighbor, the syntax is:

neighbor {*<ip-addr>* | *<peer-group-name>*} **maxas-limit in** *[<num> | disable]*

By default, neighbors or peer groups have no configured maximum values. The range is 0 – 300. The **disable** keyword is used to stop a neighbor from inheriting the configuration from the peer-group or global and to the use system default value.

To configure a peer group named “PeerGroup1” and set a maximum AS path value of 7, enter the following commands:

```
NetIron(config-bgp)# neighbor PeerGroup1 peer-group
NetIron(config-bgp)# neighbor PeerGroup1 maxas-limit in 7
```

BGP4 max-as error messages

This section lists error log messages that you might see when the router receives routes that exceed the configured AS segment limit or the internal memory limit. The log messages can contain a maximum of 30 ASNs. If a message contains more than 30 ASNs, the message is truncated and an ellipsis appears.

Maximum AS path limit error

```
SYSLOG: <11>Jan  1 00:00:00 mu1, BGP: From Peer 192.168.1.2 received Long AS_PAT
H= AS_CONFED_SET(4) 1 2 3 AS_CONFED_SEQUENCE(3) 4 AS_SET(1) 5 6 7 AS_SEQ(2) 8 9
attribute length (9) More than configured MAXAS-LIMIT 7
```

Memory limit error

```
SYSLOG: <11>Jan  1 00:00:00 mu1, BGP: From Peer 192.168.1.2 received Long AS_PAT
```

```
H= AS_CONFED_SET(4) 1 2 3 AS_CONFED_SEQUENCE(3) 4 AS_SET(1) 5 6 7 AS_SEQ(2) 8 9
attribute length (9) Exceeded internal memory limit
```

NOTE
The router generates a log message one time every two minutes. Because of this rate limit, it is possible that some errors might not appear in the log. In this case, you can use the **debug ip bgp events** command to view errors pertaining to the **maxas-limit** value and the actual AS path attributes received.

Configuring route flap dampening

Route flap dampening reduces the amount of route state changes propagated by BGP4 due to unstable routes. This in turn reduces processing requirements.

To enable route flap dampening using the default values, enter the following command.

```
NetIron(config-bgp)# dampening
```

Syntax: [**no**] **dampening** [<*half-life*> <*reuse*> <*suppress*> <*max-suppress-time*>]

The <*half-life*> parameter specifies the number of minutes after which the penalty for a route becomes half its value. The route penalty allows routes that have remained stable for a period despite earlier instability to eventually become eligible for use again. The decay rate of the penalty is proportional to the value of the penalty. After the half-life expires, the penalty decays to half its value. A dampened route that is no longer unstable can eventually again become eligible for use. You can configure the half-life to be from 1 - 45 minutes. The default is 15 minutes.

The <*reuse*> parameter specifies how low a penalty for a route must be before the route becomes eligible for use again, after being suppressed. You can set the reuse threshold to a value from 1 – 20000. The default is 750 (0.75, or three-fourths, of the penalty assessed for a one flap).

The <*suppress*> parameter specifies how high the penalty for a route can be before the device suppresses the route. You can set the suppression threshold to a value from 1 – 20000. The default is 2000 (more than two flaps).

The <*max-suppress-time*> parameter specifies the maximum number of minutes that a route can be suppressed regardless of how unstable it is. You can set the maximum suppression time to a value from 1 – 255 minutes. The default is 40 minutes.

This example shows how to change the dampening parameters.

```
NetIron(config-bgp)# dampening 20 200 2500 40
```

This command changes the half-life to 20 minutes, the reuse threshold to 200, the suppression threshold to 2500, and the maximum number of minutes a route can be dampened to 40.

NOTE
To change any of the parameters, you must specify all the parameters with the command. To want to leave any parameters unchanged, enter their default values.

Originating the default route

By default, the device does not originate and advertise a default route using BGP4. A BGP4 default route is the IP address 0.0.0.0 and the route prefix 0 or network mask 0.0.0.0. For example, 0.0.0.0/0 is a default route.

NOTE
The device checks for the existence of an IGP route for 0.0.0.0/0 in the IP route table before creating a local BGP4 route for 0.0.0.0/0.

To configure the device to originate and advertise a default BGP4 route, enter this command.

```
NetIron(config-bgp)# default-information-originate
```

Syntax: [no] **default-information-originate**

Changing the default local preference

When the device uses the BGP4 algorithm to select a route to send to the IP route table, one of the parameters the algorithm uses is the local preference. Local preference indicates a degree of preference for a route relative to other routes. BGP4 neighbors can send the local preference value as an attribute of a route in an UPDATE message.

Local preference applies only to routes within the local AS. BGP4 devices can exchange local preference information with neighbors who also are in the local AS, but BGP4 devices do not exchange local preference information with neighbors in remote ASs.

The default local preference is 100. For routes learned from EBGP neighbors, the default local preference is assigned to learned routes. For routes learned from IBGP neighbors, the local preference value is not changed for the route.

When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen.

NOTE
To set the local preference for individual routes, use route maps. Refer to "Defining route maps" on page 1068. Refer to "How BGP4 selects a path for a route" on page 1000 for information about the BGP4 algorithm.

To change the default local preference to 200, enter the following command.

```
NetIron(config-bgp)# default-local-preference 200
```

Syntax: [no] **default-local-preference** <*num*>

The <*num*> parameter indicates the preference and can be a value from 0 – 4294967295.

Changing the default metric used for redistribution

The device can redistribute directly connected routes, static IP routes, RIP routes, ISIS routes, and OSPF routes into BGP4. By default, BGP4 uses zero (0) for direct connected routes and the metric (MED) value of IGP routes in the IP route table. The MED is a global parameter that specifies the cost that will be applied to all routes, if assigned, when they are redistributed into BGP4. When routes are selected, lower metric values are preferred over higher metric values. The default, the BGP4 MED value is not assigned.

NOTE
RIP, ISIS, and OSPF also have default metric parameters. The parameters are set independently for each protocol and have different ranges.

To change the default metric to 40, enter the following command.

```
NetIron(config-bgp)# default-metric 40
```

Syntax: [**no**] **default-metric** <*value*>

The <*value*> indicates the metric and can be a value from 0 – 4294967295.

Changing the default metric used for route cost

By default, BGP4 uses the BGP MED value as the route cost when adding the route to the RTM. However, you can configure BGP4 to use the IGP cost instead.

NOTE
It is recommended that you change the default to IGP cost only in mixed-vendor environments, and that you change it on all PowerConnect and Dell devices in the environment.

To change the route cost default from BGP MED to IGP cost, enter a command such as the following:

```
NetIron(config-router-bgp)# install-igp-cost
```

Syntax: [**no**] **install-igp-cost**

Use the **no** form of the command to revert to the default of BGP MED.

Changing administrative distances

Because the device can learn about networks from various protocols, including the EBGP portion of BGP4, and IGPs such as OSPF, ISIS, and RIP, the routes to a network may differ depending on the protocol from which the routes were learned.

To select one route over another based on the source of the route information, the device can use the administrative distances assigned to the sources. The administrative distance is a protocol-independent metric that IP devices use to compare routes from different sources.

The device re-advertises a learned best BGP4 route to neighbors even when the route table manager does not also select that route for installation in the IP route table. The best BGP4 route is the BGP4 path that BGP4 selects based on comparison of the paths' BGP4 route parameters. Refer to "How BGP4 selects a path for a route" on page 1000.

When selecting a route from among different sources (BGP4, OSPF, RIP, ISIS, static routes, and so on), the software compares the routes on the basis of the administrative distance for each route. If the administrative distance of the paths is lower than the administrative distance of paths from other sources (such as static IP routes, RIP, or OSPF), the BGP4 paths are installed in the IP route table.

The default administrative distances on the device are:

- Directly connected – 0 (this value is not configurable)
- Static – 1 is the default and applies to all static routes, including default routes. This can be assigned a different value.
- EBGP – 20
- OSPF – 110
- ISIS – 115
- RIP – 120
- IBGP – 200
- Local BGP4 – 200
- Unknown – 255 (the device will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the device receives routes for the same network from OSPF and from RIP, the device will prefer the OSPF route by default. The administrative distances are configured in different places in the software. The device re-advertises a learned best BGP4 route to neighbors by default, regardless of whether the administrative distance for the route is lower than the administrative distances of other routes from different route sources to the same destination:

- To change the EBGP, IBGP, and Local BGP4 default administrative distances, refer to the instructions in this section.
- To change the default administrative distance for OSPF, RIP, ISIS, refer to "Changing administrative distances" on page 1030.
- To change the administrative distance for static routes, refer to "Configuring static routes" on page 714.

To change the default administrative distances for EBGP, IBGP, and Local BGP4, enter a command such as the following.

```
NetIron(config-bgp)# distance 200 200 200
```

Syntax: [**no**] **distance** <*external-distance*> <*internal-distance*> <*local-distance*>

The <*external-distance*> sets the EBGP distance and can be a value from 1 – 255.

The <*internal-distance*> sets the IBGP distance and can be a value from 1 – 255.

The <*local-distance*> sets the Local BGP4 distance and can be a value from 1 – 255.

Requiring the first AS to be the neighbor AS

By default, a device does not require the first AS listed in the AS_SEQUENCE field of an AS path update message from EBGP neighbors to be the AS of the neighbor that sent the update. However, you can enable the device to have this requirement. You can enable this requirement globally for the device, or for a specific neighbor or peer group. This section describes how to enable this requirement.

When you configure the device to require that the AS an EBGP neighbor is in be the same as the first AS in the AS_SEQUENCE field of an update from the neighbor, the device accepts the update only if the AS numbers match. If the AS numbers do not match, the device sends a notification message to the neighbor and closes the session. The requirement applies to all updates received from EBGP neighbors.

The hierarchy for enforcement of this feature is: a neighbor will try to use the enforce-first-as value if one is configured; if none is configured, the neighbor will try to use the configured value for a peer group. If neither configuration exists, enforcement is simply that of the global configuration (which is disabled by default).

To enable this feature globally, enter the **enforce-first-as** command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# enforce-first-as
```

Syntax: [**no**] **enforce-first-as**

To enable this feature for a specific neighbor, enter the following command at the BGP4 configuration level.

```
NetIron(config-bgp)# neighbor 1.1.1.1 enforce-first-as enable
```

Syntax: [**no**] **neighbor** <*ip-address*> **enforce-first-as** [**enable** | **disable**]

The <*ip-address*> value is the IP address of the neighbor.

When the first-as requirement is enabled, its status appears in the output of the **show running configuration** command. The optional last keyword choice of **enable** or **disable** lets you specify whether the output of the **show running configuration** command includes the configuration of the first-as requirement. This option allows the **show running configuration** command output to show what is actually configured.

To enable this feature for a peer group, enter the following command at the BGP4 configuration level.

```
NetIron(config-bgp)# neighbor Peergroup1 enforce-first-as enable
```

Syntax: [**no**] **neighbor** <*peer-group-name*> **enforce-first-as** [**enable** | **disable**]

The <*peer-group-name*> value is the name of the peer group.

When the first-as requirement is enabled, its status appears in the output of the show running configuration command. The optional last keyword choice, that of **enable** or **disable**, lets you specify whether the output of the show running configuration command includes the configuration of the first-as requirement: this option helps the show running command output to show what you have actually configured.

The following example shows a running configuration with the first-as enforcement items (for global, peer group, and neighbor) in bold.

```
NetIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
NetIron(config-bgp)# local-as 1
```

```
NetIron(config-bgp)# enforce-first-as
NetIron(config-bgp)# neighbor abc peer-group
NetIron(config-bgp)# neighbor abc remote-as 2
NetIron(config-bgp)# neighbor abc enforce-first-as disable
NetIron(config-bgp)# neighbor 192.168.1.2 peer-group abc
NetIron(config-bgp)# neighbor 192.168.1.2 enforce-first-as enable
```

Enabling fast external fallover

BGP4 devices rely on KEEPALIVE and UPDATE messages from neighbors to signify that the neighbors are alive. For BGP4 neighbors that are two or more hops away, such messages are the only indication that the BGP4 protocol has concerning the alive state of the neighbors. As a result, if a neighbor becomes non-operational, the device waits until the Hold Time expires or the TCP connection fails before concluding that the neighbor is not operational and closing its BGP4 session and TCP connection with the neighbor.

The device waits for the Hold Time to expire before ending the connection to a directly-attached BGP4 neighbor that becomes non-operational.

For directly-attached neighbors, the device immediately senses loss of a connection to the neighbor from a change of state of the port or interface that connects the device to the neighbor. For directly-attached EBGP neighbors, the device uses this information to immediately close the BGP4 session and TCP connection to locally attached neighbors that become non-operational.

NOTE
The fast external fallover feature applies only to directly attached EBGP neighbors. The feature does not apply to IBGP neighbors.

To enable fast external fallover, enter the following command.

```
NetIron(config-bgp)# fast-external-fallover
```

To disable fast external fallover again, enter the following command.

```
NetIron(config-bgp)# no fast-external-fallover
```

Syntax: [no] fast-external-fallover

Setting the local AS number

The local autonomous system number (ASN) identifies the AS in which the BGP4 device resides.

To set the local AS number, enter commands such as the following.

```
NetIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
NetIron(config-bgp)# local-as 10
NetIron(config-bgp)# write memory
```

Syntax: [no] local-as <*num*>

The <*num*> parameter specifies a local AS number in the range 1 – 4294967295. It has no default. AS numbers 64512 – 65535 are the well-known private BGP4 AS numbers and are not advertised to the Internet community.

Configuring BGP4 multipath load sharing

To change the maximum number of BGP4 shared paths, enter commands such as the following.

```
NetIron(config)# router bgp
NetIron(config-bgp)# maximum-paths 4
NetIron(config-bgp)# write memory
```

Syntax: [**no**] **maximum-paths** <*number*> | **use-load-sharing**

The <*number*> parameter specifies the maximum number of paths across which the device can balance traffic to a given BGP4 destination. You can change the maximum number of paths to a value from 2 – 8. The default is 1.

When the **use-load-sharing** option is used in place of the <*number*> variable, the maximum IP ECMP path value is determined solely by the value configured using the **ip load-sharing** command.

Customizing BGP4 multipath load sharing

By default, when BGP4 Multipath load sharing is enabled, both IBGP and EBGP paths are eligible for load sharing, while paths from different neighboring ASs are not eligible. You can change load sharing to apply only to IBGP or EBGP paths, or to support load sharing among paths from different neighboring ASs.

To enable load sharing of IBGP paths only, enter the following command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# multipath ibgp
```

To enable load sharing of EBGP paths only, enter the following command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# multipath ebgp
```

To enable load sharing of paths from different neighboring ASs, enter the following command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# multipath multi-as
```

Syntax: [**no**] **multipath ebgp** | **ibgp** | **multi-as**

The **ebgp | ibgp | multi-as** parameter specifies the change you are making to load sharing:

- **ebgp** – Load sharing applies only to EBGP paths. Load sharing is disabled for IBGP paths.
- **ibgp** – Load sharing applies only to IBGP paths. Load sharing is disabled for EBGP paths.
- **multi-as** – Load sharing is enabled for paths from different ASs.

By default, load sharing applies to EBGP and IBGP paths, and does not apply to paths from different neighboring ASs.

Enhancements to BGP4 load sharing

Enhancements to BGP4 Load Sharing allows support for load sharing of BGP4 routes in IP ECMP even if the BGP4 multipath load sharing feature is not enabled through the **use-load-sharing** option to the **maximum-paths** command. Using the following commands, you can also set separate values for IGMP and EGMP multipath load sharing.

To set the number of equal-cost multipath IBGP routes or paths that will be selected, enter commands such as the following.

```
NetIron(config)# router bgp
NetIron(config-bgp)# maximum-paths ibgp
```

Syntax: [**no**] **maximum-paths ibgp** <*num*>

The <*num*> variable specifies the number of equal-cost multipath IBGP routes that will be selected. Possible values are 1 - 8. If the value is set to 1, BGP4 level equal-cost multipath is disabled for IBGP routes.

To set the number of equal-cost multipath EBGP routes or paths that will be selected, enter commands such as the following.

```
NetIron(config)# router bgp
NetIron(config-bgp)# maximum-paths ebgp
```

Syntax: [**no**] **maximum-paths ebgp** <*num*>

The <*num*> variable specifies the number of equal-cost multipath EBGP routes that will be selected. Possible values are 1 - 8. If the value is set to 1, BGP4 level equal-cost multipath is disabled for EBGP routes.

Configuring a static BGP4 network

This feature allows you to configure a static network in BGP4, creating a stable BGP4 network in the core. While a route configured with this feature will never flap unless it is manually deleted, a “static” BGP4 network will not interrupt the normal BGP4 decision process on other learned routes being installed into the RTM (Routing Table Manager). Consequently, when there is a route that can be resolved, it will be installed into the RTM.

To configure a static BGP4 network, enter commands such as the following.

```
NetIron(config)# router bgp
NetIron(config-bgp)# static-network 209.157.22.26/16
```

Syntax: [**no**] **static-network** <*ipAddressPrefix/mask*>

The <*ipAddress/mask*> variable is the IPv4 address prefix and mask of the static BGP4 network you are creating.

Using the **no** option uninstalls a route (that was previously installed) from BGP4 RIB-IN and removes the corresponding drop route from the RTM. If there is a new best route, it is advertised to peers if necessary. Otherwise, a withdraw message is sent.

NOTE
The BGP4 network route and the BGP4 static network route are mutually exclusive. They cannot be configured with the same prefix and mask.

When you configure a route using the **static-network** command, BGP4 automatically generates a local route in BGP4 RIB-IN, and installs a NULL0 route in the RTM if there is no other valid route with the same prefix/mask learned from any peer. Otherwise, the learned BGP4 route will be installed in the RTM. In either situation, the new locally generated route will be the best route in RIB-IN and will be advertised to peers if it passes the per-peer outbound policies.

Setting an administrative distance for a static BGP4 network

When a static BGP4 network route is configured, its type is *local BGP4 route* and has a default administrative distance value of 200. To change the administrative distance value, change the value of all local BGP4 routes using the **distance** command at the router bgp level of the CLI, and set a new value for local routes as described in "Changing administrative distances" on page 1030. You can also assign a specific administrative distance value for each static network using the **distance** option as shown.

```
NetIron(config)# router bgp
NetIron(config-bgp)# static-network 209.157.22.26/16 distance 100
```

Syntax: [**no**] **static-network** <*ipAddressPrefix/mask*> **distance** <*distance-value*>

The <*ipAddress/mask*> variable is the IPv4 address prefix and mask of the static BGP4 network for which you are setting an administrative distance.

The <*distance-value*> sets the administrative distance of the static BGP4 network route. The range for this value is 1 – 255.

Limiting advertisement of a static BGP4 network to selected neighbors

You can control the advertisement of a static BGP4 network to BGP4 neighbors that are configured as Service Edge Devices. When this feature is configured for a BGP4 neighbor, static BGP4 network routes that are installed in the routing table as DROP routes are not advertised to that neighbor. When this feature is configured, the route is only advertised to identified Service Edge devices if it is installed as a forward route, such as the routes described in these steps.

1. There is a learned route from a customer BGP4 peering.
2. There is a valid learned route from another Services Edge device as a result of a customer route present on that device.

To configure a BGP4 neighbor to limit the advertisement of Static BGP4 Network routes, enter the **static-network-edge** command as shown.

```
NetIron(config)# router bgp
NetIron(config-bgp)# neighbor 1.2.3.4 static-network-edge
```

Syntax: [**no**] **neighbor** <*ip-address*> | <*peer-group-name*> **static-network-edge**

The <*ip-addr*> | <*peer-group-name*> variable indicates whether you are configuring an individual neighbor or a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group. Refer to "Configuring a BGP4 peer group" on page 1046.

Configuring paths without MEDs as the least favorable

During MED comparison, by default, the device favors a lower MED over a higher MED. Since the device assigns the value 0 to a route path MED if the MED value is missing, the default MED comparison results in the device favoring the route paths that do not have MEDs.

To configure the device to favor a route with a MED over a route that does not have a MED, enter the following command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# med-missing-as-worst
```

Syntax: [no] **med-missing-as-worst**

NOTE
This command affects route selection only when route paths are selected based on MED comparison. It is still possible for a route path without a MED to be selected based on other criteria. For example, a route path with no MED can be selected if its weight is larger than the weights of the other route paths.

Configuring BGP4 neighbors

Because BGP4 does not contain a peer discovery process, for each BGP4 neighbor (peer), you must indicate the IP address and the AS number of each neighbor. Neighbors that are in different ASs communicate using EBGP. Neighbors within the same AS communicate using IBGP.

NOTE
If the device has multiple neighbors with similar attributes, you can simplify configuration by configuring a peer group, then adding individual neighbors to it. The configuration steps are similar, except you specify a peer group name instead of a neighbor IP address when configuring the neighbor parameters, then add individual neighbors to the peer group. Refer to "Configuring a BGP4 peer group" on page 1046.

NOTE
The device attempts to establish a BGP4 session with a neighbor as soon as you enter a command specifying the IP address of the neighbor. If you want to completely configure the neighbor parameters before the device establishes a session with the neighbor, you can administratively shut down the neighbor. Refer to "Administratively shutting down a session with a BGP4 neighbor" on page 1049.

To add a BGP4 neighbor with an IP address 209.157.22.26 remote-as 100, enter the following command.

```
NetIron(config-bgp)# neighbor 209.157.22.26 remote-as 100
```

The neighbor's <*ip-addr*> must be a valid IP address.

The **neighbor** command has additional parameters, as shown in the following syntax:

Syntax: [**no**] **neighbor** {<*ip-addr*> | <*peer-group-name*>}
{
[**activate**]
[**advertisement-interval** <*seconds*>

[**allowas-in** <*num*>]
[**bfd holdover-interval** <*num*>]
[**bfd min-tx** <*num*> **min-rx** <*num*> **multiplier** <*num*>]
[**capability as4** [**enable** | **disable**]]
[**capability orf prefixlist** [**send** | **receive**]]
[**default-originate** [**route-map** <*map-name*>]]
[**description** <*string*>]
[**distribute-list in** | **out** <*num,num,...*> | <*acl-num*> **localin** | **out**]
[**ebgp-btsh**]
[**ebgp-multihop** <*num*>]
[**enforce-first-as**]
[**filter-list** <*access-list-name*> [**in** | **out**]]
[**local-as** <*as-num*> [**no-prepend**]]
[**maxas-limit in** [<*num*> |**disable**]
[**maximum-prefix** <*num*> [<*threshold*>] [**teardown**]
[**next-hop-self**]
[**password** <*string*>]
[**peer-group** <*group-name*>]
[**prefix-list** <*string*> **in** | **out**]
[**remote-as** <*as-number*>]
[**remove-private-as**]
[**route-map in** | **out** <*map-name*>]
[**route-reflector-client**]
[**send-community**]
[**shutdown** [**generate-rib-out**]]
[**soft-reconfiguration inbound**]
[**static-network-edge**]
[**timers keep-alive** <*num*> **hold-time** <*num*>]
[**unsuppress-map** <*map-name*>]
[**update-source** <*ip-addr*> | **ethernet** <*slot*>/<*portnum*> | **pos** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*>]
[**weight** <*num*>]
}

The <*ip-addr*> | <*peer-group-name*> parameter indicates whether you are configuring an individual neighbor or a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group. Refer to "Configuring a BGP4 peer group" on page 1046.

activate allows exchange of routes in the current family mode.

advertisement-interval <*seconds*> configures an interval in seconds over which the specified neighbor or peer group will hold all route updates before sending them. At the expiration of the timer, the routes are sent as a batch. The default value for this parameter is zero. Acceptable values are 0 to 600 seconds.

allowas-in <*num*> disables the AS_PATH check function for routes learned from a specified location. BGP4 usually rejects routes that contain an AS number within an AS_PATH attribute to prevent routing loops. In an MPLS or VPN hub and spoke topology this can prevent legitimate routes from being accepted. The **allowas-in** option stops this blockage. <*num*> specifies the number of occurrences of the AS number.

bfd holdover-interval and **bfd min-tx** options are described in "Configuring BFD for BGP4" on page 2156.

capability as4 [**enable** | **disable**] enables the capability of processing AS4s. The optional keywords **enable** | **disable** specify whether the feature should be changed from its current state. For example, if this neighbor belongs to a peer group that is enabled for AS4s but you want disable it on the current interface, use the command and include the **disable** keyword.

capability orf prefixlist [**send** | **receive**] configures cooperative device filtering. The **send** | **receive** parameter specifies the support you are enabling:

- **send** – The device sends the IP prefix lists as Outbound Route Filters (ORFs) to the neighbor.
- **receive** – The device accepts filters as Outbound Route Filters (ORFs) from the neighbor.

If you do not specify either **send** or **receive**, both capabilities are enabled. The **prefixlist** parameter specifies the type of filter you want to send to the neighbor.

For more information, refer to "Configuring cooperative BGP4 route filtering" on page 1083.

NOTE
The current release supports cooperative filtering only for filters configured using IP prefix lists.

default-originate [**route-map** <*map-name*>] configures the device to send the default route 0.0.0.0 to the neighbor. If you use the route-map <*map-name*> parameter, the route map injects the default route conditionally, based on the match conditions in the route map.

description <*string*> specifies a name for the neighbor. You can enter an alphanumeric text string up to 80 characters long.

distribute-list in | out <*num,num,...*> specifies a distribute list to be applied to updates to or from the specified neighbor. The **in** | **out** keywords specify whether the list is applied on updates received from the neighbor, or sent to the neighbor. The <*num,num,...*> parameter specifies the list of address-list filters. The device applies the filters in the order in which you list them and stops applying the filters in the distribute list when a match is found.

To use an IP ACL instead of a distribute list, you can specify **distribute-list** <*acl-num*> **in** | **out** . In this case, <*acl-num*> is an IP ACL.

NOTE
By default, if a route does not match any of the filters, the device denies the route. To change the default behavior, configure the last filter as **permit any any**.

NOTE
The address filter must already be configured. Refer to "Defining and applying IP prefix lists" on page 1067.

ebgp-btsh enables GTSM protection for the specified neighbor. For details, see "Generalized TTL Security Mechanism support" on page 1103.

ebgp-multihop [<*num*>] specifies that the neighbor is more than one hop away and that the session type with the neighbor is EBGP-multihop. This option is disabled by default. The <*num*> parameter specifies the TTL you are adding for the neighbor. You can specify a number from 0 – 255. The default is 0. If you leave the EBGP TTL value set to 0, the software uses the IP TTL value.

enforce-first-as ensures, for this neighbor, that the first AS listed in the AS_SEQUENCE field of an AS path update message from EBGP neighbors is the AS of the neighbor that sent the update. For details, refer to "Requiring the first AS to be the neighbor AS" on page 1032.

filter-list in | out <*num,num,...*> specifies an AS-path filter list or a list of AS-path ACLs. The **in | out** keywords specify whether the list is applied on updates received from the neighbor or sent to the neighbor. If you specify **in** or **out**, The <*num,num,...*> parameter specifies the list of AS-path filters. The device applies the filters in the order in which you list them and stops applying the filters in the AS-path filter list when a match is found. The **weight** <*num*> parameter specifies a weight that the device applies to routes received from the neighbor that match the AS-path filter or ACL. You can specify a number from 0 – 65535.

Alternatively, you can specify **filter-list** <*acl-num*> **in | out | weight** to use an AS-path ACL instead of an AS-path filter list. In this case, <*acl-num*> is an AS-path ACL.

NOTE
By default, if an AS-path does not match any of the filters or ACLs, the device denies the route. To change the default behavior, configure the last filter or ACL as **permit any any**.

NOTE
The AS-path filter or ACL must already be configured. Refer to "Filtering AS-paths" on page 1063.

local-as <*as-num*> assigns a local AS number with the value specified by the <*as-num*> variable to the neighbor being configured. The <*as-number*> has no default value. Its range is 1 – 4294967295.

NOTE
When the **local-as** option is used, the device automatically prepends the local AS number to the routes that are received from the EBGP peer; to disable this behavior, include the **no-prepend** keyword.

maxas-limit in <*num*> |**disable** specifies that the router discard routes that exceed a maximum AS path length received in UPDATE messages. You can specify a value from 0 – 300. The default value is 300. The **disable** keyword is used to stop a neighbor from inheriting the configuration from the peer-group or global and to the use system default value.

maximum-prefix <*num*> specifies the maximum number of IP network prefixes (routes) that can be learned from the specified neighbor or peer group. You can specify a value from 0 – 4294967295. The default is 0 (unlimited).

- The <*num*> value is the maximum number. The range is 0 – 4294967295. The default is 0 (unlimited).
- The <*threshold*> parameter specifies the percentage of the value you specified for the **maximum-prefix** <*num*>, at which you want the software to generate a Syslog message. You can specify a value from 1 (one percent) to 100 (100 percent). The default is 100.
- The **teardown** parameter tears down the neighbor session if the maximum-prefix limit is exceeded. The session remains shutdown until you clear the prefixes using the **clear ip bgp neighbor all** or **clear ip bgp neighbor** <*ip-addr*> command, or change the maximum-prefix configuration for the neighbor. The software also generates a Syslog message.

next-hop-self specifies that the device should list itself as the next hop in updates sent to the specified neighbor. This option is disabled by default.

password <*string*> specifies an MD5 password for securing sessions between the device and its neighbor. You can enter a string up to 80 characters long. The string can contain any alphanumeric characters and spaces if the words in the password are placed inside quotes.

The system creates an MD5 hash of the password and use it for securing sessions between the device and its neighbors. To display the configuration, the system uses a 2-way encoding scheme to be able to retrieve the original password that was entered.

By default, password is encrypted. If you want password to be in clear text, insert a **0** between **password** and <*string*>.

```
NetIron(config-bgp)# neighbor 209.157.22.26 remote-as password 0 marmalade
```

The system adds an encryption code followed by the encrypted text of the original password. For example, the following portion of the code has the encrypted code "2".

```
password 2 $IUA2PWc9LW9VIW9zVQ=="
```

One of the following may be displayed:

- 0 = the password is not encrypted and is in clear text
- 1 = the password uses proprietary simple crytographic 2-way algorithm
- 2 = the password uses proprietary base64 crytographic 2-way algorithm

peer-group <*group-name*> assigns the neighbor to the specified peer group.

prefix-list <*string*> **in** | **out** specifies an IP prefix list. You can use IP prefix lists to control routes to and from the neighbor. IP prefix lists are an alternative method to AS-path filters. The **in** | **out** keywords specify whether the list is applied on updates received from the neighbor or sent to the neighbor. The filters can use the same prefix list or different prefix lists. To configure an IP prefix list, refer to "Defining and applying IP prefix lists" on page 1067.

remote-as <*as-number*> specifies the AS in which the remote neighbor resides. The <*as-number*> has no default value. The range is 1 – 4294967295.

remove-private-as configures the device to remove private AS numbers from update messages the device sends to this neighbor. The device will remove AS numbers 64512 – 65535 (the well-known BGP4 private AS numbers) from the AS-path attribute in update messages the device sends to the neighbor. This option is disabled by default.

route-map in | **out** <*map-name*> specifies a route map the device will apply to updates sent to or received from the specified neighbor. The **in** | **out** keywords specify whether the list is applied on updates received from the neighbor or sent to the neighbor.

NOTE
The route map must already be configured. Refer to """Defining route maps" on page 1068.

route-reflector-client specifies that this neighbor is a route-reflector client of the device. Use the parameter only if this device is going to be a route reflector. For information, refer to "Configuring a route reflector" on page 1016. This option is disabled by default.

send-community enables sending the community attribute in updates to the specified neighbor. By default, the device does not send the community attribute.

shutdown administratively shuts down the session with this neighbor. Shutting down the session lets you configure the neighbor and save the configuration without actually establishing a session with the neighbor.

When a peer is put into the shutdown state, ribout routes are not produced for that peer. You can elect to produce ribout routes using the **generate-rib-out** option. This option is disabled by default.

soft-reconfiguration inbound enables the soft reconfiguration feature, which stores all the route updates received from the neighbor. If you request a soft reset of inbound routes, the software performs the reset by comparing the policies against the stored route updates, instead of requesting the neighbor BGP4 route table or resetting the session with the neighbor. Refer to "Using soft reconfiguration" on page 1090.

static-network-edge controls the advertisement of a static BGP4 network to BGP4 neighbors that are configured as Service Edge Devices. For more information, refer to "Limiting advertisement of a static BGP4 network to selected neighbors" on page 1036.

timers keep-alive <*num*> **hold-time** <*num*> overrides the global settings for the Keep Alive Time and Hold Time. For the Keep Alive Time, you can specify 0 – 65535 seconds. For the Hold Time, you can specify 0 or a number in the range 3 – 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the device waits indefinitely for messages from a neighbor without concluding that the neighbor is non-operational. The defaults for these parameters are the currently configured global Keep Alive Time and Hold Time. For more information about these parameters, refer to "Changing the Keep Alive Time and Hold Time" on page 1057.

unsuppress-map <*map-name*> removes route suppression from neighbor routes when those routes have been suppressed due to aggregation. Refer to "Removing route dampening from suppressed routes" on page 1043.

update-source <*ip-addr*> | **ethernet** <*slot*>/<*portnum*> | **pos** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> configures the device to communicate with the neighbor through the specified interface. There is no default.

weight <*num*> specifies a weight the device will add to routes received from the specified neighbor. BGP4 prefers larger weights over smaller weights. The default weight is 0.

Auto shutdown of BGP4 neighbors on initial configuration

You can use the global **auto-shutdown-new-neighbors** command within the **router bgp** configuration to disable establishment of the BGP4 connection with a remote peer when the peer is first configured, as follows.

```
NetIron(config)# router bgp
NetIron(config-bgp)# auto-shutdown-new-neighbors
```

Once all of the configuration parameters for the peer are complete, you can start the BGP4 session establishment process using the **no** option with the existing peer shutdown option command to disable the peer shutdown state.

```
NetIron(config)# router bgp
NetIron(config-bgp)# no neighbor 1.1.1.1 shutdown
```

Syntax: [**no**] **neighbor** <*ip-address*> **shutdown**

If auto shutdown of BGP4 neighbors is enabled and you want to disable it to allow a new BGP4 peer configured to establish a connection with remote peers, use the **no** option with the command.

```
NetIron(config)# router bgp
NetIron(config-bgp)# no auto-shutdown-new-neighbors
```

Syntax: [**no**] **auto-shutdown-new-neighbors**

The default state for auto shutdown of BGP4 neighbors is disabled.

NOTE
When the **auto-shutdown-new-neighbors** value is changed, the value of the shutdown parameter for any of the existing configured neighbors is not changed. Any new BGP4 neighbor configured after the setting of the **auto-shutdown-new-neighbors** command will have the shutdown state set to the current value of the **auto-shutdown-new-neighbors** command. Previously configured peer group parameters are not affected by the **auto-shutdown-new-neighbors** command.

When a new peer group is created and new neighbors belonging to this peer group are being configured, you can use the **peer group shutdown** parameter to prevent the establishment of connections with remote peers.

Using the auto shutdown of BGP4 neighbors during a configuration

To control when a newly-configured BGP4 neighbor establishes a BGP4 session, use the **auto-shutdown-new-neighbors** command.

```
NetIron(config)# router bgp
NetIron(config-bgp)# auto-shutdown-new-neighbors
NetIron(config-bgp)# neighbor 1.1.1.1 remote-as 200
```

Once all of the BGP4 neighbor configuration parameters are configured, use the following commands to establish the BGP4 session.

```
NetIron(config)# router bgp
NetIron(config-bgp)# no neighbor 1.1.1.1 shutdown
```

Removing route dampening from suppressed routes

You can selectively unsuppress specific routes that have been suppressed due to aggregation, and allow these routes to be advertised to a specific neighbor or peer group.

```
NetIron(config-bgp)# aggregate-address 209.1.0.0 255.255.0.0 summary-only
NetIron(config-bgp)# show ip bgp route 209.1.0.0/16 longer
Number of BGP Routes matching display condition : 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Prefix             Next Hop        Metric     LocPrf     Weight Status
1      209.1.0.0/16       0.0.0.0                   101        32768  BAL
AS_PATH:
2      209.1.44.0/24      10.2.0.1        1         101        32768  BLS
AS_PATH:
```

In this example, the **aggregate-address** command configures an aggregate address of 209.1.0.0 255.255.0.0. and the **summary-only** parameter prevents the device from advertising more specific routes contained within the aggregate route.

Entering a **show ip bgp route** command for the aggregate address 209.1.0.0/16 shows that the more specific routes aggregated into 209.1.0.0/16 have been suppressed. In this case, the route to 209.1.44.0/24 has been suppressed. If you enter this command, the display shows that the route is not being advertised to the BGP4 neighbors.

```
NetIron(config-bgp)# show ip bgp route 209.1.44.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Prefix             Next Hop        Metric     LocPrf     Weight Status
1      209.1.44.0/24      10.2.0.1        1          101        32768  BLS
AS_PATH:
Route is not advertised to any peers
```

To override the **summary-only** parameter and allow a specific route to be advertised to a neighbor, enter commands such as the following

```
NetIron(config)# ip prefix-list Unsuppress1 permit 209.1.44.0/24
NetIron(config)# route-map RouteMap1 permit 1
NetIron(config-routemap RouteMap1)# match prefix-list Unsuppress1
NetIron(config-routemap RouteMap1)# exit
NetIron(config)# router bgp
NetIron(config-bgp)# neighbor 10.1.0.2 unsuppress-map RouteMap1
NetIron(config-bgp)# clear ip bgp neighbor 10.1.0.2 soft-out
```

The **ip prefix-list** command configures an IP prefix list for network 209.1.44.0/24, which is the route you want to unsuppress. The next two commands configure a route map that uses the prefix list as input. The **neighbor** command enables the device to advertise the routes specified in the route map to neighbor 10.1.0.2. The **clear** command performs a soft reset of the session with the neighbor so that the device can advertise the unsuppressed route.

Syntax: [**no**] **neighbor** <*ip-addr*> | <*peer-group-name*> **unsuppress-map** <*map-name*>

The **show ip bgp route** command verifies that the route has been unsuppressed.

```
NetIron(config-bgp)# show ip bgp route 209.1.44.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Prefix             Next Hop        Metric     LocPrf     Weight Status
1      209.1.44.0/24      10.2.0.1        1          101        32768  BLS
AS_PATH:
Route is advertised to 1 peers:
10.1.0.2(4)
```

Encrypting BGP4 MD5 authentication keys

When you configure a BGP4 neighbor or neighbor peer group, you can specify an MD5 authentication string to authenticate packets exchanged with the neighbor or peer group of neighbors.

For added security, by default, the software encrypts the display of the authentication string. The software also provides an optional parameter to disable encryption of the authentication string, on an individual neighbor or peer group basis. By default, MD5 authentication strings are displayed in encrypted format in the output of the following commands:

- **show running-config** (or **write terminal**)
- **show configuration**
- **show ip bgp config**

When encryption of the authentication string is enabled, the string is encrypted in the CLI regardless of the access level you are using.

When you save the configuration to the startup configuration file, the file contains the new BGP4 command syntax and encrypted passwords or strings.

NOTE
It is recommended that you save a copy of the startup configuration file for each device you plan to upgrade.

Encryption example

The following commands configure a BGP4 neighbor and a peer group, and specify MD5 authentication strings (passwords) to authenticate packets exchanged with the neighbor or peer group.

```
NetIron(config-bgp)# local-as 2
NetIron(config-bgp)# neighbor xyz peer-group
NetIron(config-bgp)# neighbor xyz password abc
NetIron(config-bgp)# neighbor 10.10.200.102 peer-group xyz
NetIron(config-bgp)# neighbor 10.10.200.102 password test
```

The BGP4 configuration commands appear in the following format as a result of the **show ip bgp configuration** command.

```
NetIron(config-bgp)# show ip bgp configuration
Current BGP configuration:
router bgp
 local-as 2
 neighbor xyz peer-group
 neighbor xyz password 2 $b24tbw==
 neighbor 10.10.200.102 peer-group xyz
 neighbor 10.10.200.102 remote-as 1
 neighbor 10.10.200.102 password 2 $on-o
```

In this output, the software has converted the commands that specify an authentication string into the new syntax (described below), and has encrypted display of the authentication strings.

Syntax: [**no**] **neighbor** <*ip-addr*> |<*peer-group-name*> **password** <*string*>

The <*ip-addr*> | <*peer-group-name*> parameters indicate whether you are configuring an individual neighbor or a peer group. If you specify the IP address of a neighbor, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group.

The password <*string*> parameter specifies an MD5 authentication string to secure sessions between the device and the neighbor. You can enter a string of up to 80 characters. The string can contain any alphanumeric characters, but must be placed inside quotes if it contains a space.

The system creates an MD5 hash of the password and uses it to secure sessions between the device and the neighbors. To display the configuration, the system uses a 2-way encoding scheme to retrieve the original password.

By default, password is encrypted. If you want the password to be in clear text, insert a **0** between **password** and <*string*>.

```
NetIron(config-bgp)# neighbor 209.157.22.26 remote-as password 0 admin
```

The system adds an encryption code followed by the encrypted text of the original password. For example, the following portion of the code has the encrypted code “2”.

```
password 2 $IUA2PWc9LW9VIW9zVQ=="

NetIron(config-bgp)# neighbor 209.157.22.26 remote-as password 0 marmalade
```

One of the following may be displayed:

- 0 - the password is not encrypted and is in clear text
- 1 - the password uses proprietary simple crytographic 2-way algorithm
- 2 - the password uses proprietary base64 crytographic 2-way algorithm

Displaying the authentication string

To display the authentication string, enter the following commands.

```
NetIron(config)# enable password-display
NetIron(config)# show ip bgp neighbors
```

The **enable password-display** command enables display of the authentication string, but only in the output of the **show ip bgp neighbors** command. String display is still encrypted in the startup configuration file and running configuration.

Configuring a BGP4 peer group

A ***peer group*** is a set of BGP4 neighbors that share common parameters. The benefits of peer groups are:

- **Simplified neighbor configuration** – You can configure a set of neighbor parameters and then apply them to multiple neighbors. You do not need to configure the common parameters individually on each neighbor.
- **Flash memory conservation** – Using peer groups instead of individually configuring all the parameters for each neighbor requires fewer configuration commands in the startup configuration file.

You can perform the following tasks on a peer-group basis:

- Reset neighbor sessions
- Perform soft-outbound resets (the device updates outgoing route information to neighbors but does not entirely reset the sessions with those neighbors)
- Clear BGP4 message statistics
- Clear error buffers

Peer group parameters

You can set all neighbor parameters in a peer group. When you add a neighbor to the peer group, the neighbor receives all the parameter settings you set in the group, except parameter values you have explicitly configured for the neighbor. If you do not set a neighbor parameter in the peer group and the parameter also is not set for the individual neighbor, the neighbor uses the default value.

Configuration rules

The following rules apply to peer group configuration.

- You must configure a peer group before you can add neighbors to the peer group.
- If you remove a parameter from a peer group, the value for that parameter is reset to the default for all the neighbors within the peer group, unless you have explicitly set that parameter on individual neighbors. In this case, the value you set on the individual neighbors applies to those neighbors, while the default value applies to neighbors for which you have not explicitly set the value.

NOTE
If you enter a command to remove the remote AS parameter from a peer group, the software makes sure that the peer group does not contain any neighbors. If the peer group contains neighbors, the software does not allow you to remove the remote AS so that the neighbors in the peer group that are using the remote AS do not lose connectivity to the device.

You can override neighbor parameters on an individual neighbor basis:

- If you do not specify a parameter for an individual neighbor, the neighbor uses the value in the peer group.
- If you set the parameter for the individual neighbor, that value overrides the value you set in the peer group.
- If you add a parameter to a peer group that already contains neighbors, the parameter value is applied to neighbors that do not already have the parameter explicitly set. If a neighbor has the parameter explicitly set, the explicitly set value overrides the value you set for the peer group.
- If you remove the setting for a parameter from a peer group, the value for that parameter changes to the default value for all the neighbors in the peer group that do not have that parameter individually set.

Configuring a peer group

To configure a peer group, enter commands such as the following at the BGP4 configuration level.

```
NetIron(config-bgp)# neighbor PeerGroup1 peer-group
NetIron(config-bgp)# neighbor PeerGroup1 description “EastCoast Neighbors”
NetIron(config-bgp)# neighbor PeerGroup1 remote-as 100
NetIron(config-bgp)# neighbor PeerGroup1 distribute-list out 1
NetIron(config-bgp)# neighbor PeerGroup1 capability as4
```

The commands in this example configure a peer group called “PeerGroup1” and set the following parameters for the peer group:

- A description, “EastCoast Neighbors”
- A remote AS number, 100
- A distribute list for outbound traffic
- The capability of Peergroup1 to utilize four-byte AS number

The software applies these parameters to each neighbor you add to the peer group. You can override the description parameter for individual neighbors. If you set the description parameter for an individual neighbor, the description overrides the description configured for the peer group.

Syntax: **neighbor** <*peer-group-name*> **peer-group**

The <*peer-group-name*> parameter specifies the name of the group and can be up to 80 characters long. The name can contain special characters and internal blanks. If you use internal blanks, you must use quotation marks around the name. For example, the command **neighbor "My Three Peers"** peer-group is valid, but the command **neighbor My Three Peers** peer-group is not valid.

Syntax: [**no**] **neighbor** <*ip-addr*> |<*peer-group-name*>
[**default-originate** [**route-map** <*map-name*>]]
[**description** <*string*>]
[**distribute-list in** | **out** <*num,num,...*> | <*acl-num*> **in** | **out**]
[**ebgp-multihop** [<*num*>]]
[**filter-list in** | **out** <*num,num,...*> | <*acl-num*> **in** | **out** | **weight**]
[**maxas-limit in** [<*num*> |**disable**]
[**maximum-prefix** <*num*> [<*threshold*>] [**teardown**]]
[**next-hop-self**]
[**password** <*string*>]
[**prefix-list** <*string*> **in** | **out**]
[**remote-as** <*as-number*>]
[**remove-private-as**]
[**route-map in** | **out** <*map-name*>]
[**route-reflector-client**]
[**send-community**]
[**soft-reconfiguration inbound**]
[**shutdown**]
[**timers keep-alive** <*num*> **hold-time** <*num*>]
[**update-source loopback** <*num*> **ethernet** <*slot*>/<*portnum*> | **pos** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*>]
[**weight** <*num*>]
[**local-as** <*as-num*>]

The <*ip-addr*> |<*peer-group-name*> parameters indicate whether you are configuring a peer group or an individual neighbor. You can specify a peer group name or IP address with the **neighbor** command. If you specify a peer group name, you are configuring a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. Use the <*ip-addr*> parameter if you are configuring an individual neighbor instead of a peer group. Refer to "Configuring BGP4 neighbors" on page 1037 and "Configuring a BGP4 peer group" on page 1046.

The remaining parameters are the same ones supported for individual neighbors. Refer to "Configuring BGP4 neighbors" on page 1037 and "Configuring a BGP4 peer group" on page 1046.

Applying a peer group to a neighbor

After you configure a peer group, you can add neighbors to the group. When you add a neighbor to a peer group, you are applying all the neighbor attributes specified in the peer group to the neighbor.

To add neighbors to a peer group, enter commands such as the following.

```
NetIron(config-bgp)# neighbor 192.168.1.12 peer-group PeerGroup1
NetIron(config-bgp)# neighbor 192.168.2.45 peer-group PeerGroup1
NetIron(config-bgp)# neighbor 192.168.3.69 peer-group PeerGroup1
```

The commands in this example add three neighbors to the peer group “PeerGroup1”. As members of the peer group, the neighbors automatically receive the neighbor parameter values configured for the peer group. You also can override the parameters on an individual neighbor basis. For neighbor parameters not specified for the peer group, the neighbors use the default values.

Syntax: [**no**] **neighbor** <*ip-addr*> **peer-group** <*peer-group-name*>

The <*ip-addr*> parameter specifies the IP address of the neighbor.

The <*peer-group-name*> parameter specifies the peer group name.

NOTE
You must add the peer group before you can add neighbors to it.

Administratively shutting down a session with a BGP4 neighbor

You can prevent the device from starting a BGP4 session with a neighbor by administratively shutting down the neighbor. This option is very useful for situations in which you want to configure parameters for a neighbor, but are not ready to use the neighbor. You can shut the neighbor down as soon as you have added it the device, configure the neighbor parameters, then allow the device to reestablish a session with the neighbor by removing the shutdown option from the neighbor.

When you apply the option to shut down a neighbor, the option takes place immediately and remains in effect until you remove it. If you save the configuration to the startup configuration file, the shutdown option remains in effect even after a software reload.

The software also contains an option to end the session with a BGP4 neighbor and clear the routes learned from the neighbor. Unlike this clear option, the option for shutting down the neighbor can be saved in the startup configuration file and can prevent the device from establishing a BGP4 session with the neighbor even after reloading the software.

NOTE
If you notice that a particular BGP4 neighbor never establishes a session with the device, check the running configuration and startup configuration files for that device to see whether the configuration contains a command that is shutting down the neighbor. The neighbor may have been shut down previously by an administrator.

To shut down a BGP4 neighbor, enter commands such as the following.

```
NetIron(config)# router bgp
NetIron(config-bgp)# neighbor 209.157.22.26 shutdown
NetIron(config-bgp)# write memory
```

Syntax: [**no**] **neighbor** <*ip-addr*> **shutdown**

The <*ip-addr*> parameter specifies the IP address of the neighbor.

Specifying a list of networks to advertise

By default, the device sends BGP4 routes only for the networks you either identify with the **network** command or are redistributed into BGP4 from OSPF, ISIS, RIP, or connected routes.

NOTE
The exact route must exist in the IP route table before the device can create a local BGP4 route.

To configure the device to advertise network 209.157.22.0/24, enter the following command.

```
NetIron(config-bgp)# network 209.157.22.0 255.255.255.0
```

Syntax: [**no**] **network** <*ip-addr*> <*ip-mask*> [**route-map** <*map-name*>] | [**weight** <*num*>] | [**backdoor**]

The <*ip-addr*> is the network number and the <*ip-mask*> specifies the network mask.

The **route-map** <*map-name*> parameter specifies the name of the route map you want to use to set or change BGP4 attributes for the network you are advertising. The route map must already be configured. If it is not, the default action is to deny redistribution.

The **weight** <*num*> parameter specifies a weight to be added to routes to this network.

The **backdoor** parameter changes the administrative distance of the route to this network from the EBGP administrative distance (20 by default) to the Local BGP4 weight (200 by default), tagging the route as a backdoor route. Use this parameter when you want the device to prefer IGP routes such as RIP or OSPF routes over the EBGP route for the network.

Specifying a route map when configuring BGP4 network advertising

You can specify a route map when you configure a BGP4 network to be advertised. The device uses the route map to set or change BGP4 attributes when creating a local BGP4 route.

NOTE
You must configure the route map *before* you can specify the route map name in a BGP4 network configuration; otherwise, the route is not imported into BGP4.

To configure a route map, and use it to set or change route attributes for a network you define for BGP4 to advertise, enter commands such as the following.

```
NetIron(config)# route-map set_net permit 1
NetIron(config-routemap set_net)# set community no-export
NetIron(config-routemap set_net)# exit
NetIron(config)# router bgp
NetIron(config-bgp)# network 100.100.1.0/24 route-map set_net
```

The first two commands in this example create a route map named “set_net” that sets the community attribute for routes that use the route map to “NO_EXPORT”. The next two commands change the CLI to the BGP4 configuration level. The last command configures a network for advertising from BGP4, and associates the “set_net” route map with the network. When BGP4 originates the 100.100.1.0/24 network, BGP4 also sets the community attribute for the network to “NO_EXPORT”.

Syntax: [**no**] **network** <*ip-addr*> <*ip-mask*> [**route-map** <*map-name*>] | [**weight** <*num*>] | [**backdoor**]

The **route-map** <*map-name*> parameter specifies the name of the route map you want to use to set or change BGP4 attributes for the network you are advertising. The route map must already be configured.

For information about the other parameters, refer to “Defining route maps” on page 1068.

Using the IP default route as a valid next-hop for a BGP4 route

By default, the device does not use a default route to resolve a BGP4 next-hop route. If the IP route lookup for the BGP4 next-hop does not result in a valid IGP route (including static or direct routes), the BGP4 next-hop is considered to be unreachable and the BGP4 route is not used.

In some cases, such as when the device is acting as an edge device, you can allow the device to use the default route as a valid next-hop. To do so, enter the following command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# next-hop-enable-default
```

Syntax: [no] next-hop-enable-default

Enabling next-hop recursion

For each BGP4 route learned, the device performs a route lookup to obtain the IP address of the next-hop for the route. A BGP4 route is eligible for addition in the IP route table only if the following conditions are true:

- The lookup succeeds in obtaining a valid next-hop IP address for the route.
- The path to the next-hop IP address is an IGP path or a static route path.

By default, the software performs only one lookup for the next-hop IP address for the BGP4 route. If the next-hop lookup does not result in a valid next-hop IP address, or the path to the next-hop IP address is a BGP4 path, the software considers the BGP4 route destination to be unreachable. The route is not eligible to be added to the IP route table.

The BGP4 route table can contain a route with a next-hop IP address that is not reachable through an IGP route, even though the device can reach a hop farther away through an IGP route. This can occur when the IGPs do not learn a complete set of IGP routes, so the device learns about an internal route through IBGP instead of through an IGP. In this case, the IP route table will not contain a route that can be used to reach the BGP4 route destination.

To enable the device to find the IGP route to the next-hop gateway for a BGP4 route, enable recursive next-hop lookups. With this feature enabled, if the first lookup for a BGP4 route results in an IBGP path that originated within the same AS, rather than an IGP path or static route path, the device performs a lookup on the next-hop IP address for the next-hop gateway. If this second lookup results in an IGP path, the software considers the BGP4 route to be valid and adds it to the IP route table. Otherwise, the device performs another lookup on the next-hop IP address of the next-hop for the next-hop gateway, and so on, until one of the lookups results in an IGP route.

NOTE
You must configure a static route or use an IGP to learn the route to the EBGP multihop peer.

Example when recursive route lookups are disabled

The output here shows the results of an unsuccessful next-hop lookup for a BGP4 route. In this case, next-hop recursive lookups are disabled. This example is for the BGP4 route to network 240.0.0.0/24.

```
NetIron# show ip bgp route
Total number of BGP Routes: 5
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix             Next Hop        Metric     LocPrf      Weight Status
1      0.0.0.0/0           10.1.0.2       0          100         0      BI
         AS_PATH: 65001 4355 701 80
2      102.0.0.0/24        10.0.0.1       1          100         0      BI
         AS_PATH: 65001 4355 1
3      104.0.0.0/24        10.1.0.2       0          100         0      BI
         AS_PATH: 65001 4355 701 1 189
4      240.0.0.0/24        102.0.0.1      1          100         0      I
         AS_PATH: 65001 4355 3356 7170 1455
5      250.0.0.0/24        209.157.24.1   1          100         0      I
         AS_PATH: 65001 4355 701
```

In this example, the device cannot reach 240.0.0.0/24, because the next-hop IP address for the route is an IBGP route instead of an IGP route, and is considered unreachable by the device. The IP route table entry for the next-hop gateway for the BGP4 route's next-hop gateway (102.0.0.1/24) is shown here.

```
NetIron# show ip route 102.0.0.1
Total number of IP routes: 37
      Network Address   Gateway          Port    Cost   Type
      102.0.0.0         10.0.0.1         1/1     1      B
```

Since the route to the next-hop gateway is a BGP4 route, and not an IGP route, it cannot be used to reach 240.0.0.0/24. In this case, the device tries to use the default route, if present, to reach the subnet that contains the BGP4 route next-hop gateway.

```
NetIron# show ip route 240.0.0.0/24
Total number of IP routes: 37
      Network Address   Gateway          Port    Cost   Type
      0.0.0.0           10.0.0.202       1/1     1      S
```

Example when recursive route lookups are enabled

When recursive next-hop lookups are enabled, the device continues to look up the next-hop gateways along the route until the device finds an IGP route to the BGP4 route destination.

```
NetIron# show ip bgp route
Total number of BGP Routes: 5
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix         Next Hop        Metric     LocPrf      Weight Status
1      0.0.0.0/0       10.1.0.2       0          100         0      BI
         AS_PATH: 65001 4355 701 80
2      102.0.0.0/24    10.0.0.1       1          100         0      BI
         AS_PATH: 65001 4355 1
3      104.0.0.0/24    10.1.0.2       0          100         0      BI
         AS_PATH: 65001 4355 701 1 189
4      240.0.0.0/24   102.0.0.1       1          100         0      BI
         AS_PATH: 65001 4355 3356 7170 1455
5      250.0.0.0/24   209.157.24.1    1          100         0       I
         AS_PATH: 65001 4355 701
```

The first lookup results in an IBGP route, to network 102.0.0.0/24.

```
NetIron# show ip route 102.0.0.1
Total number of IP routes: 38
      Network Address   Gateway           Port    Cost   Type
      102.0.0.0         10.0.0.1          1/1     1      B
         AS_PATH: 65001 4355 1
```

Since the route to 102.0.0.1/24 is not an IGP route, the device cannot reach the next hop through IP, and so cannot use the BGP4 route. In this case, since recursive next-hop lookups are enabled, the device next performs a lookup for the next-hop gateway to 102.0.0.1's next-hop gateway, 10.0.0.1.

```
NetIron# show ip bgp route 102.0.0.0
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix         Next Hop        Metric     LocPrf      Weight Status
1      102.0.0.0/24   10.0.0.1        1          100        0      BI
         AS_PATH: 65001 4355 1
```

The next-hop IP address for 102.0.0.1 is not an IGP route, which means the BGP4 route destination still cannot be reached through IP. The recursive next-hop lookup feature performs a lookup on the next-hop gateway for 10.0.0.1

```
NetIron# show ip route 10.0.0.1
Total number of IP routes: 38
      Network Address   Gateway           Port    Cost   Type
      10.0.0.0          0.0.0.0           1/1     1      D
        AS_PATH: 65001 4355 1
```

This lookup results in an IGP route that is a directly-connected route. As a result, the BGP4 route destination is now reachable through IGP, which means the BGP4 route can be added to the IP route table. The IP route table with the BGP4 route is shown here.

```
NetIron# show ip route 240.0.0.0/24
Total number of IP routes: 38
      Network Address    Gateway           Port    Cost   Type
      240.0.0.0          10.0.0.1          1/1     1      B
         AS_PATH: 65001 4355 1
```

The device can use this route because it has an IP route to the next-hop gateway. Without recursive next-hop lookups, this route would not be in the IP route table.

Enabling recursive next-hop lookups

The recursive next-hop lookups feature is disabled by default.

To enable recursive next-hop lookups, enter the following command at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# next-hop-recursion
```

Syntax: [no] **next-hop-recursion**

Modifying redistribution parameters

By default, the device does not redistribute route information between BGP4 and the IP IGPs (RIP, ISIS, and OSPF). You can configure the device to redistribute OSPF, ISIS, or RIP routes, directly connected routes, or static routes into BGP4.

To enable redistribution of all OSPF routes and directly attached routes into BGP4, enter the following commands.

```
NetIron(config)# router bgp
NetIron(config-bgp)# redistribute ospf
NetIron(config-bgp)# redistribute connected
NetIron(config-bgp)# write memory
```

Syntax: [no] **redistribute connected** | **ospf** | **rip** | **isis** | **static**

The **connected** parameter indicates that you are redistributing routes to directly attached devices into BGP4.

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

NOTE
Entering **redistribute ospf** simply redistributes internal OSPF routes. To redistribute external OSPF routes also, use the **redistribute ospf match external...** command. Refer to "Redistributing OSPF external routes" on page 1055.

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **isis** parameter indicates that you are redistributing ISIS routes into BGP4.

The **static** parameter indicates that you are redistributing static routes into BGP4.

Redistributing connected routes

To configure BGP4 to redistribute directly connected routes, enter the following command.

```
NetIron(config-bgp)# redistribute connected
```

Syntax: [**no**] **redistribute connected** [**metric** <*num*>] [**route-map** <*map-name*>]

The **connected** parameter indicates that you are redistributing routes to directly attached devices into BGP4.

The **metric** <*num*> parameter changes the metric. You can specify a value from 0 – 4294967295. The default is not assigned.

The **route-map** <*map-name*> parameter specifies a route map to be consulted before adding the RIP route to the BGP4 route table.

NOTE
The route map you specify must already be configured on the device. Refer to "Defining route maps" on page 1068 for information about defining route maps.

Redistributing RIP routes

To configure BGP4 to redistribute RIP routes and add a metric of 10 to the redistributed routes, enter the following command.

```
NetIron(config-bgp)# redistribute rip metric 10
```

Syntax: [**no**] **redistribute rip** [**metric** <*num*>] [**route-map** <*map-name*>]

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **metric** <*num*> parameter changes the metric. You can specify a value from 0 – 4294967295. The default is not assigned.

The **route-map** <*map-name*> parameter specifies a route map to be consulted before adding the RIP route to the BGP4 route table.

NOTE
The route map you specify must already be configured on the device. Refer to "Defining route maps" on page 1068 for information about defining route maps.

Redistributing OSPF external routes

To configure the device to redistribute OSPF external type 1 routes, enter the following command.

```
NetIron(config-bgp)# redistribute ospf match external1
```

Syntax: [**no**] **redistribute ospf** [**match internal** | **external1** | **external2**] [**metric** <*num*>] [**route-map** <*map-name*>]

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

The match **internal | external1 | external2** parameters apply only to OSPF. These parameters specify the types of OSPF routes to be redistributed into BGP4. The default is internal.

NOTE
If you do not enter a value for the **match** parameter, (for example, you enter r**edistribute ospf** only) then only internal OSPF routes will be redistributed.

The **metric** <*num*> parameter changes the metric. You can specify a value from 0 – 4294967295. The default is not assigned.

The **route-map** <*map-name*> parameter specifies a route map to be consulted before adding the OSPF route to the BGP4 route table.

NOTE
The route map you specify must already be configured on the device. Refer to "Defining route maps" on page 1068 for information about defining route maps.

NOTE
If you use both the **redistribute ospf route-map** <*map-name*> command and the **redistribute ospf match internal** | **external1** | **external2** command, the software uses only the route map for filtering.

Redistributing IS-IS

To configure the device to redistribute IS-IS routes, enter the following command.

```
NetIron(config-bgp)# redistribute isis level-1
```

Syntax: [**no**] **redistribute isis level-1** | **level-1-2** | **level-2** [**metric** <*num*>] [**route-map** <*map-name*>]

The **isis** parameter indicates that you are redistributing IS-IS routes into BGP4.

The **level-1** parameter redistributes IS-IS routes only within the area the routes.

The **level-2** parameter redistributes IS-IS routes between areas within a domain.

The **level-1-2** parameter redistributes IS-IS routes within the area of the routes and between areas within a domain.

The **metric** <*num*> parameter changes the metric. You can specify a value from 0 – 4294967295. The default is not assigned.

The **route-map** <*map-name*> parameter specifies a route map to be consulted before adding the OSPF route to the BGP4 route table.

Redistributing static routes

To configure the device to redistribute static routes, enter the following command.

```
NetIron(config-bgp)# redistribute static
```

Syntax: [**no**] **redistribute static** [**metric** <*num*>] [**route-map** <*map-name*>]

The **static** parameter indicates that you are redistributing static routes into BGP4.

The **metric** <*num*> parameter changes the metric. You can specify a value from 0 – 4294967295. The default is 0.

The **route-map***p-name*> parameter specifies a route map to be consulted before adding the static route to the BGP4 route table.

NOTE
The route map you specify must already be configured on the device. Refer to "Defining route maps" on page 1068 for information about defining route maps.

Using a table map to set the tag value

Route maps that contain set statements change values in routes when the routes are accepted by the route map. For inbound route maps (route maps that filter routes received from neighbors), the routes are changed before they enter the BGP4 route table.

For tag values, if you do not want the value to change until a route enters the IP route table, you can use a table map to change the value. A table map is a route map that you have associated with the IP routing table. The device applies the set statements for tag values in the table map to routes before adding them to the route table.

To configure a table map, you first configure the route map, then identify it as a table map. The table map does not require separate configuration. You can have one table map.

NOTE
Use table maps only for setting the tag value. Do not use table maps to set other attributes. To set other route attributes, use route maps or filters.

To create a route map and identify it as a table map, enter commands such as following. These commands create a route map that uses an address filter. For routes that match the IP prefix list filter, the route map changes the tag value to 100 and is then considered as a table map. This route map is applied only to routes the device places in the IP route table. The route map is not applied to all routes. This example assumes that IP prefix list p11 has already been configured.

```
NetIron(config)# route-map TAG_IP permit 1
NetIron(config-routemap TAG_IP)# match ip address prefix-list p11
NetIron(config-routemap TAG_IP)# set tag 100
NetIron(config-routemap TAG_IP)# router bgp
NetIron(config-bgp)# table-map TAG_IP
```

Changing the Keep Alive Time and Hold Time

The Keep Alive Time specifies how frequently the device will send KEEPALIVE messages to its BGP4 neighbors. The Hold Time specifies how long the device will wait for a KEEPALIVE or UPDATE message from a neighbor before concluding that the neighbor is dead. When the device concludes that a BGP4 neighbor is dead, the device ends the BGP4 session and closes the TCP connection to the neighbor.

The default Keep Alive time is 60 seconds. The default Hold Time is 180 seconds.

NOTE
Generally, you should set the Hold Time to three times the value of the Keep Alive Time.

NOTE
You can override the global Keep Alive Time and Hold Time on individual neighbors. Refer to "Configuring BGP4 neighbors" on page 1037 and "Configuring a BGP4 peer group" on page 1046.

To change the Keep Alive Time to 30 and Hold Time to 90, enter the following command.

```
NetIron(config-bgp)# timers keep-alive 30 hold-time 90
```

Syntax: **[no] timers keep-alive** <*num*> **hold-time** <*num*>

For each keyword, <*num*> indicates the number of seconds. The Keep Alive Time can be 0 – 65535. The Hold Time can be 0 or 3 – 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the device waits indefinitely for messages from a neighbor without concluding that the neighbor is dead.

Changing the BGP4 next-hop update timer

By default, the device updates the BGP4 next-hop tables and affected BGP4 routes five seconds after IGP route changes. You can change the update timer to a value from 1 – 30 seconds.

To change the BGP4 update timer value to 15 seconds, for example, enter a command such as the following at the BGP4 configuration level of the CLI.

```
NetIron(config-bgp)# update-time 15
```

Syntax: [**no**] **update-time** <*secs*>

The <*secs*> parameter specifies the number of seconds and can be from 0 – 30. The default is 5. The value of 0 permits fast BGP4 convergence for situations such as link-failure or IGP route changes. Setting the value to 0 starts the BGP4 route calculation in sub-second time. All other values from 1 to 30 are still calculated in seconds

Changing the device ID

The OSPF and BGP4 protocols use device IDs to identify devices that are running the protocols. A device ID is a valid, unique IP address and sometimes is an IP address configured on the device. The device ID cannot be an IP address in use by another device.

By default, the device ID on a device is one of the following:

- If the device has loopback interfaces, the default device ID is the IP address on the lowest numbered loopback interface configured on the device. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default device ID is 9.9.9.9/24:
  - Loopback interface 1, 9.9.9.9/24
  - Loopback interface 2, 4.4.4.4/24
  - Loopback interface 3, 1.1.1.1/24
- If the device does not have any loopback interfaces, the default device ID is the lowest numbered IP interface address configured on the device.

NOTE
A device uses the same device ID for both OSPF and BGP4. If the device is already configured for OSPF, you may want to use the device ID that already assigned to the device rather than set a new one. To display the current device ID, enter the **show ip** CLI command at any CLI level.

To change the device ID, enter a command such as the following.

```
NetIron(config)# ip router-id 209.157.22.26
```

Syntax: [**no**] **ip router-id** <*ip-addr*>

The <*ip-addr*> can be any valid, unique IP address.

NOTE
You can specify an IP address used for an interface on the device, but do not specify an IP address that is being used by another device.

Adding a loopback interface

You can configure the device to use a loopback interface instead of a specific port or virtual routing interface to communicate with a BGP4 neighbor. A loopback interface adds stability to the network by working around route flap problems that can occur due to unstable links between the device and neighbors.

Loopback interfaces are always up, regardless of the states of physical interfaces. Loopback interfaces are especially useful for IBGP neighbors (neighbors in the same AS) that are multiple hops away from the device. When you configure a BGP4 neighbor on the device, you can specify whether the device uses the loopback interface to communicate with the neighbor. As long as a path exists between the device and the neighbor, BGP4 information can be exchanged. The BGP4 session is not associated with a specific link, but is instead associated with the virtual interfaces.

NOTE
If you configure the device to use a loopback interface to communicate with a BGP4 neighbor, the peer IP address on the remote device pointing to your loopback address must be configured.

To add a loopback interface, enter commands such as the following.

```
NetIron(config-bgp)# exit
NetIron(config)# int loopback 1
NetIron(config-lbif-1)# ip address 10.0.0.1/24
```

Syntax: [**no**] **interface loopback** <*num*>

The <*num*> value can be from 1 – 64.

Changing the maximum number of paths for BGP4 load sharing

Load sharing enables the device to balance traffic to a route across multiple equal-cost paths of the same route type (EBGP or IBGP).

To configure the device to perform BGP4 load sharing:

- Enable IP load sharing if it is disabled.
- Set the maximum number of BGP4 load sharing paths. The default maximum number is 1, which means no BGP4 load sharing takes place by default. Refer to "Configuring BGP4 multipath load sharing" on page 1034.

NOTE
The maximum number of BGP4 load sharing paths cannot be greater than the maximum number of IP load sharing paths.

How load sharing affects route selection

During evaluation of multiple paths to select the best path to a given destination (for installment in the IP route table), the device performs a final comparison of the internal paths. The following events occur when load sharing is enabled or disabled:

- When load sharing is disabled, the device prefers the path with the lower device ID if the **compare-routerid** command is enabled.
- When load sharing and BGP4 load sharing are enabled, the device balances the traffic across multiple paths instead of choosing just one path based on device ID.

Refer to "How BGP4 selects a path for a route" on page 1000 for a description of the BGP4 algorithm.

When you enable IP load sharing, the device can load-balance BGP4 or OSPF routes across up to four equal paths by default. You can change the number load sharing paths to a value from 2 – 8.

Configuring route reflection parameters

Normally, all the BGP4 devices within an AS are fully meshed. Since each device has an IBGP session with each of the other BGP4 devices in the AS, each device has a route for each IBGP neighbor. For large ASs containing many IBGP devices, the IBGP route information in each fully-meshed IBGP device may introduce too much administrative overhead.

To avoid this overhead, you can organize your IGP devices into clusters:

- A ***cluster*** is a group of IGP devices organized into route reflectors and route reflector clients. You configure the cluster by assigning a cluster ID on the route reflector and identifying the IGP neighbors that are members of that cluster. All configuration for route reflection takes place on the route reflectors. Clients are unaware that they are members of a route reflection cluster. All members of the cluster must be in the same AS. The cluster ID can be any number from 1 – 4294967295, or an IP address. The default is the device ID.

  NOTE
  If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops within the cluster.

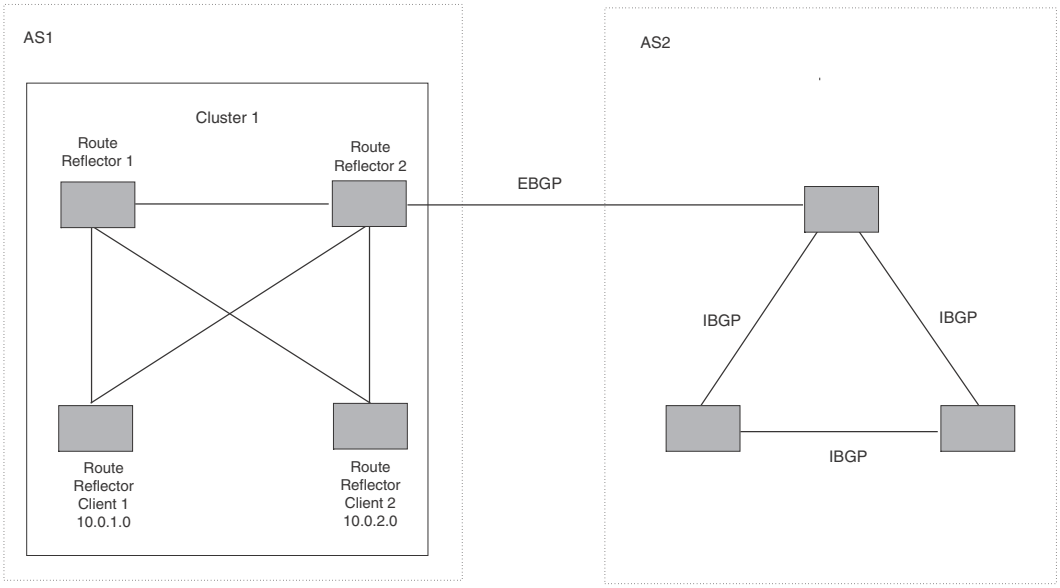
- A ***route reflector*** is an IGP device configured to send BGP4 route information to all the clients (other BGP4 devices) within the cluster. Route reflection is enabled on all BGP4 devices by default but does not take effect unless you add route reflector clients to the device.
- A ***route reflector client*** is an IGP device identified as a member of a cluster. You identify a device as a route reflector client on the device that is the route reflector, not on the client. The client itself requires no additional configuration. In fact, the client does not know that it is a route reflector client. The client just knows that it receives updates from its neighbors and does not know whether one or more of those neighbors are route reflectors.

NOTE
Route reflection applies only among IBGP devices within the same AS. You cannot configure a cluster that spans multiple ASs.

Figure 26.4 shows an example of a route reflector configuration. In this example, two devices are configured as route reflectors for the same cluster, which provides redundancy in case one of the reflectors becomes unavailable. Without redundancy, if a route reflector becomes unavailable, the clients for that router are cut off from BGP4 updates.

AS1 contains a cluster with two route reflectors and two clients. The route reflectors are fully meshed with other BGP4 devices, but the clients are not fully meshed and rely on the route reflectors to propagate BGP4 route updates.

FIGURE 153 A route reflector configuration



Support for RFC 4456

Route reflection is based on RFC 4456. This updated RFC helps eliminate routing loops that are possible in some implementations of the older specification, RFC 1966. These instances include:

- The device adds the route reflection attributes only if it is a route reflector, and only when advertising IBGP route information to other IBGP neighbors. The attributes are not used when communicating with EBGP neighbors.
- A device configured as a route reflector sets the ORIGINATOR_ID attribute to the device ID of the device that originated the route. The route reflector sets this attribute only if this is the first time the route is being reflected (sent by a route reflector).
- If a device receives a route with an ORIGINATOR_ID attribute value that is the same as the ID of the device, the device discards the route and does not advertise it. By discarding the route, the device prevents a routing loop.
- The first time a route is reflected by a device configured as a route reflector, the route reflector adds the CLUSTER_LIST attribute to the route. Other route reflectors that receive the route from an IBGP neighbor add their cluster IDs to the front of the routes CLUSTER_LIST. If the route reflector does not have a cluster ID configured, the device adds its device ID to the front of the CLUSTER_LIST.
- If a device configured as a route reflector receives a route with a CLUSTER_LIST that contains the cluster ID of the route reflector, the route reflector discards the route.

Configuration procedures

NOTE
All configuration for route reflection takes place on the route reflectors, not on the clients.

Enter the following commands to configure a device as route reflector 1 in "Route filters used by each protocol" on page 1080. To configure route reflector 2, enter the same commands on the device that will be route reflector 2. The clients require no configuration for route reflection.

```
NetIron(config-bgp)# cluster-id 1
```

Syntax: [**no**] **cluster-id** <*num*> | <*ip-addr*>

The <*num*> | <*ip-addr*> parameters specify the cluster ID and can be a number from 1 – 4294967295, or an IP address. The default is the device ID. You can configure one cluster ID on the device. All route-reflector clients for the device are members of the cluster.

NOTE
If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops in the cluster.

To add an IBGP neighbor to the cluster, enter the following command:

```
NetIron(config-bgp)# neighbor 10.0.1.0 route-reflector-client
```

Syntax: [**no**] **neighbor** <*ip-addr*> **route-reflector-client**

For more information about the **neighbor** command, refer to "Configuring BGP4 neighbors" on page 1037 and "Configuring a BGP4 peer group" on page 1046.

Filtering

This section describes how to configure filters for AS-paths, communities, and other BGP4 attributes.

Filtering AS-paths

You can filter updates received from BGP4 neighbors based on the contents of the AS-path list accompanying the updates. For example, to deny routes that have the AS 4.3.2.1 in the AS-path from entering the BGP4 route table, you can define a filter.

The device provides the following methods for filtering on AS-path information:

- AS-path filters - refer to "Setting the local AS number" on page 1033.
- AS-path ACLs

NOTE
The device cannot support AS-path filters and AS-path ACLs at the same time. Use one method or the other, but do not mix methods.

NOTE
Once you define a filter or ACL, the default action for updates that do not match a filter is **deny**. To change the default action to **permit**, configure the last filter or ACL as **permit any any**.

AS-path filters or AS-path ACLs can be referred to by the filter list number of a BGP4 neighbor as well as by match clauses in a route map.

Defining an AS-path ACL

To configure an AS-path list that uses ACL 1, enter a command such as the following.

```
NetIron(config)# ip as-path access-list acl1 permit 100
NetIron(config)# router bgp
NetIron(config-bgp)# neighbor 10.10.10.1 filter-list 1 in
```

Syntax: [**no**] **ip as-path access-list** <*string*> [**seq** <*seq-value*>] **deny** | **permit** <*regular-expression*>

The **ip as-path** command configures an AS-path ACL that permits routes containing AS number 100 in their AS paths. The **neighbor** command then applies the AS-path ACL to advertisements and updates received from neighbor 10.10.10.1. In this example, the only routes the device permits from neighbor 10.10.10.1 are those whose AS-paths contain AS-path number 100.

The <*string*> parameter specifies the ACL name. (If you enter a number, the CLI interprets the number as a text string.)

The **seq** <*seq-value*> parameter is optional and specifies the sequence number for the AS-path list. If you do not specify a sequence number, the software numbers in increments of 5, beginning with number 5. The software interprets the entries in an AS-path list in numerical order, beginning with the lowest sequence number.

The **deny** | **permit** parameter specifies the action the software takes if the AS-path list for a route matches a match clause in this ACL. To configure the AS-path match clauses in a route map, use the match as-path command. Refer to "Matching based on AS-path ACL" on page 1071.

The <*regular-expression*> parameter specifies the AS path information you want to permit or deny to routes that match any of the match clauses within the ACL. You can enter a specific AS number or use a regular expression.

The **neighbor** command uses the **filter-list** parameter to apply the AS-path ACL to the neighbor. Refer to "Configuring BGP4 neighbors" on page 1037 and "Configuring a BGP4 peer group" on page 1046.

Using regular expressions

Use a regular expression for the <*as-path*> parameter to specify a single character or multiple characters as a filter pattern. If the AS-path matches the pattern specified in the regular expression, the filter evaluation is true; otherwise, the evaluation is false.

You can also include special characters that influence the way the software matches the AS-path against the filter value.

To filter on a specific single-character value, enter the character for the <*as-path*> parameter. For example, to filter on AS-paths that contain the letter "z", enter the following command:

```
NetIron(config-bgp)# ip as-path access-list acl1 permit z
```

To filter on a string of multiple characters, enter the characters in brackets. For example, to filter on AS-paths that contain "x", "y", or "z", enter the following command.

```
NetIron(config-bgp)# ip as-path access-list acl1 permit [xyz]
```

Special characters

When you enter a single-character expression or a list of characters, you also can use the special characters listed in Table 167. The description for each character includes an example. Some special characters must be placed in front of the characters they control and others must be placed after the characters they control. The examples show where to place the special character.

TABLE 167 BGP4 special characters for regular expressions

Character	Operation
.	The period matches on any single character, including a blank space. For example, the following regular expression matches for "aa", "ab", "ac", and so on, but not just "a". a.
*	The asterisk matches on zero or more sequences of a pattern. For example, the following regular expression matches on an AS-path that contains the string "1111" followed by any value: 1111*
+	The plus sign matches on one or more sequences of a pattern. For example, the following regular expression matches on an AS-path that contains a sequence of "g"s, such as "deg", "degg", "deggg", and so on: deg+
?	The question mark matches on zero occurrences or one occurrence of a pattern. For example, the following regular expression matches on an AS-path that contains "dg" or "deg": de?g
^	A caret (when not used within brackets) matches on the beginning of an input string. For example, the following regular expression matches on an AS-path that begins with "3": ^3

TABLE 167 BGP4 special characters for regular expressions (Continued)

Character	Operation
$	A dollar sign matches on the end of an input string. For example, the following regular expression matches on an AS-path that ends with “deg”: deg$
_	An underscore matches on one or more of the following: • , (comma) • { (left curly brace) • } (right curly brace) • ((left parenthesis) •) (right parenthesis) • The beginning of the input string • The end of the input string • A blank space For example, the following regular expression matches on “100” but not on “1002”, “2100”, and so on. _100_
[ ]	Square brackets enclose a range of single-character patterns. For example, the following regular expression matches on an AS-path that contains “1”, “2”, “3”, “4”, or “5”: [1-5] You can use the following expression symbols within the brackets. These symbols are allowed only inside the brackets: • ^ – The caret matches on any characters except the ones in the brackets. For example, the following regular expression matches on an AS-path that does not contain “1”, “2”, “3”, “4”, or “5”: [^1-5] • - The hyphen separates the beginning and ending of a range of characters. A match occurs if any of the characters within the range is present. Refer to the example above.
|	A vertical bar (sometimes called a pipe or a “logical or”) separates two alternative values or sets of values. The AS-path can match one or the other value. For example, the following regular expression matches on an AS-path that contains either “abc” or “defg”: (abc)|(defg) **NOTE:** The parentheses group multiple characters to be treated as one value. Refer to the following row for more information about parentheses.
( )	Parentheses allow you to create complex expressions. For example, the following complex expression matches on “abc”, “abcabc”, or “abcabcabcdefg”, but not on “abcdefgdefg”: ((abc)+)|((defg)?)

To filter for a special character instead of using the special character as described in Table 167, enter “\” (backslash) in front of the character. For example, to filter on AS-path strings containing an asterisk, enter the asterisk portion of the regular expression as “*”.

```
NetIron(config-bgp)# ip as-path access-list acl2 deny \*
```

To use the backslash as a string character, enter two slashes. For example, to filter on AS-path strings containing a backslash, enter the backslash portion of the regular expression as “\\”.

```
NetIron(config-bgp)# ip as-path access-list acl2 deny \\
```

Filtering communities

You can filter routes received from BGP4 neighbors based on community names.

A community is an optional attribute that identifies the route as a member of a user-defined class of routes. Community names are arbitrary values made of two five-digit integers joined by a colon. You determine what the name means when you create the community name as a route attribute. Each string in the community name can be a number from 0 – 65535.

This format allows you to easily classify community names. For example, a common convention used in community naming is to configure the first string as the local AS and the second string as the unique community within that AS. Using this convention, communities 1:10, 1:20, and 1:30 can be easily identified as member communities of AS 1.

The device provides the following methods for filtering on community information.

- Community filters - refer to "Filtering communities" on page 1065.
- Community list ACLs

NOTE
The device cannot actively support community filters and community list ACLs at the same time. Use one method or the other but do not mix methods.

NOTE
Once you define a filter or ACL, the default action for communities that do not match a filter or ACL is **deny**. To change the default action to **permit**, configure the last filter or ACL entry as **permit any any**.

Community filters or ACLs can be referred to by match clauses in a route map.

Defining a community ACL

To configure community ACL 1, enter a command such as the following. This command configures a community ACL that permits routes that contain community 123:2.

NOTE
Refer to "Matching based on community ACL" on page 1072 for information about how to use a community list as a match condition in a route map.

```
NetIron(config)# ip community-list 1 permit 123:2
```

Syntax: [**no**] **ip community-list standard** <*string*> [**seq** <seq-value>] **deny** | **permit** <*community-num*>

Syntax: [**no**] **ip community-list extended** <*string*> [**seq** <*seq-value*>] **deny** | **permit** <*community-num*> | <*regular-expression*>

The <*string*> parameter specifies the ACL name. (If you enter a number, the CLI interprets the number as a text string.)

The **standard** or **extended** parameter specifies whether you are configuring a standard or extended community ACL. The difference between standard and extended communities is that a standard community ACL does not support regular expressions and an extended one does.

The **seq** <*seq-value*> parameter is optional and specifies the sequence number for the community list. You can configure up to 199 entries in a community list. If you do not specify a sequence number, the software numbers the entries in increments of 5, beginning with number 5. The software interprets the entries in a community list in numerical order, beginning with the lowest sequence number.

The **deny** | **permit** parameters specify the action the software takes if a route community list matches a match clause in this ACL. To configure the community-list match clauses in a route map, use the **match community** command. Refer to "Matching based on community ACL" on page 1072.

The <*community-num*> parameter specifies the community type or community number. This parameter can have the following values:

- <*num*>:<*num*> – A specific community number
- **internet** – The Internet community
- **no-export** – The community of sub-ASs within a confederation. Routes with this community can be exported to other sub-ASs within the same confederation but cannot be exported outside the confederation to other ASs or otherwise sent to EBGP neighbors.
- **local-as** – The local sub-AS within the confederation. Routes with this community can be advertised only within the local subAS.
- **no-advertise** – Routes with this community cannot be advertised to any other BGP4 devices at all.

The <*regular-expression*> parameter specifies a regular expression for matching on community names. For information about regular expression syntax, refer to "Using regular expressions" on page 1064. You can specify a regular expression only in an extended community ACL.

To use a community-list filter, use route maps with the **match community** parameter.

Defining and applying IP prefix lists

An IP prefix list specifies a list of networks. When you apply an IP prefix list to a neighbor, the device sends or receives only a route whose destination is in the IP prefix list. The software interprets the prefix lists in order, beginning with the lowest sequence number.

To configure an IP prefix list and apply it to a neighbor, enter commands such as the following.

```
NetIron(config)# ip prefix-list Routesfor20 permit 20.20.0.0/24
NetIron(config)# router bgp
NetIron(config-bgp)# neighbor 10.10.10.1 prefix-list Routesfor20 out
```

These commands configure an IP prefix list named Routesfor20, which permits routes to network 20.20.0.0/24. The **neighbor** command configures the device to use IP prefix list Routesfor20 to determine which routes to send to neighbor 10.10.10.1. The device sends routes that go to 20.20.x.x to neighbor 10.10.10.1 because the IP prefix list explicitly permits these routes to be sent to the neighbor.

Syntax: [**no**] **ip prefix-list** <*name*> [**seq** <*seq-value*>] [**description** <*string*>] **deny** | **permit** <*network-addr*>/<*mask-bits*> [**ge** <*ge-value*>] [**le** <*le-value*>]

The <*name*> parameter specifies the prefix list name. Use this name when applying the prefix list to a neighbor.

The **description** <*string*> parameter is a text string describing the prefix list.

The **seq** <*seq-value*> parameter is optional and specifies the sequence number of the IP prefix list. If you do not specify a sequence number, the software numbers the entries in increments of 5, beginning with prefix list entry 5. The software interprets the prefix list entries in numerical order, beginning with the lowest sequence number.

The **deny | permit** parameter specifies the action the software takes if a neighbor route is in this prefix list.

The <*network-addr*>/<*mask-bits*> parameters specify the network number and the number of bits in the network mask.

You can specify a range of prefix length for prefixes that are more specific than <*network-addr*>/<*mask-bits*>.

The prefix-list matches only on this network unless you use the **ge** <*ge-value*> or **le** <*le-value*> parameters.

- If you specify only **ge** <*ge-value*>, the mask-length range is from <*ge-value*> to 32.
- If you specify only **le** <*le-value*>, the mask-length range is from length to <*le-value*>.

The <*ge-value*> or <*le-value*> you specify must meet the following condition:

length < ge-value <= le-value <= 32

If you do not specify **ge** <*ge-value*> or **le** <*le-value*>, the prefix list matches only on the exact network prefix you specified with the <*network-addr*>/<*mask-bits*> parameter.

For the syntax of the **neighbor** command shown in this example, refer to "Configuring BGP4 neighbors" on page 1037 and "Configuring a BGP4 peer group" on page 1046.

Defining neighbor distribute lists

A neighbor distribute list is a list of BGP4 address filters or ACLs that filter the traffic to or from a neighbor.

To configure a distribute list that uses ACL 1, enter a command such as the following.

```
NetIron(config-bgp)# neighbor 10.10.10.1 distribute-list 1 in
```

This command configures the device to use ACL 1 to select the routes that the device will accept from neighbor 10.10.10.1.

Syntax: [**no**] **neighbor** <*ip-addr*> **distribute-list** <*name-or-num*> **in** | **out**

The <*ip-addr*> parameter specifies the neighbor.

The <*name-or-num*> parameter specifies the name or number of a standard, extended, or named ACL.

The **in** | **out** parameters specify whether the distribute list applies to inbound or outbound routes:

- **in** – controls the routes the device will accept from the neighbor.
- **out** – controls the routes sent to the neighbor.

Defining route maps

A ***route map*** is a named set of match conditions and parameter settings that the device can use to modify route attributes and to control redistribution of the routes into other protocols. A route map consists of a sequence of ***instances***. If you think of a route map as a table, an instance is a row in that table. The device evaluates a route according to route map instances in ascending numerical order. The route is first compared against instance 1, then against instance 2, and so on. When a match is found, the device stops evaluating the route.

Route maps can contain ***match clauses*** and ***set*** statements. Each route map contains a **permit** or **deny** action for routes that match the match clauses:

- If the route map contains a **permit** action, a route that matches a match statement is permitted; otherwise, the route is denied.
- If the route map contains a **deny** action, a route that matches a match statement is denied.
- If a route does not match any match statements in the route map, the route is denied. This is the default action. To change the default action, configure the last match statement in the last instance of the route map to **permit any any**.
- If there is no match statement, the software considers the route to be a match.
- For route maps that contain address filters, AS-path filters, or community filters, if the action specified by a filter conflicts with the action specified by the route map, the route map action takes precedence over the filter action.

If the route map contains set clauses, routes that are permitted by the route map match statements are modified according to the set clauses.

Match statements compare the route against one or more of the following:

- The route BGP4 MED (metric)
- A sequence of AS-path filters
- A sequence of community filters
- A sequence of address filters
- The IP address of the next hop device
- The route tag
- For OSPF routes only, the route type (internal, external type-1, or external type-2)
- An AS-path ACL
- A community ACL
- An IP prefix list
- An IP ACL

For routes that match all of the match statements, the route map set clauses can perform one or more of the following modifications to the route attributes:

- Prepend AS numbers to the front of the route AS-path. By adding AS numbers to the AS-path, you can cause the route to be less preferred when compared to other routes based on the length of the AS-path.
- Add a user-defined tag an automatically calculated tag to the route.
- Set the community value.
- Set the local preference.
- Set the MED (metric).
- Set the IP address of the next-hop device.
- Set the origin to IGP or INCOMPLETE.
- Set the weight.
- Set a BGP4 static network route.

When you configure parameters for redistributing routes into BGP4, one of the optional parameters is a route map. If you specify a route map as one of the redistribution parameters, the device matches the route against the match statements in the route map. If a match is found and if the route map contains set clauses, the device sets the attributes in the route according to the set clauses.

To create a route map, you define instances of the map by a sequence number.

To define a route map, use the procedures in the following sections.

Entering the route map into the software

To add instance 1 of a route map named "GET_ONE" with a permit action, enter the following command.

```
NetIron(config)# route-map GET_ONE permit 1
NetIron(config-routemap GET_ONE)#
```

Syntax: [**no**] **route-map** <*map-name*> **permit** | **deny** <*num*>

As shown in this example, the command prompt changes to the route map level. You can enter the match and set clauses at this level. Refer to "Specifying the match conditions" on page 1070 and "Setting parameters in the routes" on page 1075.

The <*map-name*> is a string of characters that names the map. Map names can be up to 32 characters in length.

The **permit** | **deny** parameter specifies the action the device will take if a route matches a match statement:

- If you specify **deny**, the device does not advertise or learn the route.
- If you specify **permit**, the device applies the match and set clauses associated with this route map instance.

The <*num*> parameter specifies the instance of the route map you are defining.

To delete a route map, enter a command such as the following. When you delete a route map, all the permit and deny entries in the route map are deleted.

```
NetIron(config)# no route-map Map1
```

This command deletes a route map named Map1. All entries in the route map are deleted.

To delete a specific instance of a route map without deleting the rest of the route map, enter a command such as the following.

```
NetIron(config)# no route-map Map1 permit 10
```

This command deletes the specified instance from the route map but leaves the other instances of the route map intact.

Specifying the match conditions

Use the following command to define the match conditions for instance 1 of the route map GET_ONE. This instance compares the route updates against BGP4 address filter 11.

```
NetIron(config-routemap GET_ONE)# match address-filters 11
```

Syntax: [**no**] **match**
[**as-path** <*name*>]
[**community** <*acl*> **exact-match**] |
[**ip address** <*acl*> | **prefix-list** <*string*>] |
[**ip route-source** <*acl*> | **prefix** <*name*>]
[**metric** <*num*>] |
[**next-hop** <*address-filter-list*>] |

[**route-type internal** | **external-type1** | **external-type2**] | [**level-1** | **level-2** | **level-1-2**]
[**tag** <tag-value>] |
interface <*interface*> <*interface*> <*interface*> ..
protocol bgp static-network
protocol bgp external
protocol bgp internal

The **as-path** <*num*> parameter specifies an AS-path ACL. You can specify up to five AS-path ACLs. To configure an AS-path ACL, use the **ip as-path access-list** command. Refer to "Defining an AS-path ACL" on page 1063.

The **community** <*num*> parameter specifies a community ACL.

NOTE
The ACL must already be configured.

The **community** <*acl*> **exact-match** parameter matches a route if (and only if) the route community attributes field contains the same community numbers specified in the match statement.

The **ip address** | **next-hop** <*acl-num*> | **prefix-list** <*string*> parameters specify an ACL or IP prefix list. Use this parameter to match based on the destination network or next-hop gateway. To configure an IP ACL for use with this command, use the **ip access-list** command. Refer to the Access Control List chapter. To configure an IP prefix list, use the **ip prefix-list** command.

The **ip route-source** <*acl*> | **prefix** <*name*> parameters match based on the source of a route (the IP address of the neighbor from which the device learned the route).

The **metric** <*num*> parameter compares the route MED (metric) to the specified value.

The **next-hop** <*address-filter-list*> parameter compares the IP address of the route next-hop to the specified IP address filters. The filters must already be configured.

The **route-type internal** | **external-type1 | external-type2** parameters apply only to OSPF routes. These parameters compare the route type to the specified value. The **level-1** parameter compares IS-IS routes only with routes within the same area. The **level-2** parameter compares IS-IS routes only with routes in different areas, but within a domain. The **level-1-2** parameter compares IS-IS routes with routes in the same area and in different areas, but within a domain.

The **tag** <*tag-value*> parameter compares the route tag to the specified tag value.

The **protocol bgp static-network** parameter matches on BGP4 static network routes.

The **protocol bgp external** parameter matches on eBGP (external) routes.

The **protocol bgp internal** parameter matches on iBGP (internal) routes.

The following sections contain examples of how to configure route maps that include match statements that match on ACLs.

Matching based on AS-path ACL

To construct a route map that matches based on AS-path ACL 1, enter the following commands.

```
NetIron(config)# route-map PathMap permit 1
NetIron(config-routemap PathMap)# match as-path 1
```

Syntax: [**no**] **match as-path** <*num*>

The <*num*> parameter specifies an AS-path ACL and can be a number from 1 – 199. You can specify up to five AS-path ACLs. To configure an AS-path ACL, use the **ip as-path access-list** command. Refer to "Defining an AS-path ACL" on page 1063.

Matching based on community ACL

To construct a route map that matches based on community ACL 1, enter the following commands.

```
NetIron(config)# ip community-list 1 permit 123:2
NetIron(config)# route-map CommMap permit 1
NetIron(config-routemap CommMap)# match community 1
```

Syntax: [**no**] **match community** <*string*>

The <*string*> parameter specifies a community list ACL. To configure a community list ACL, use the **ip community-list** command. Refer to "Defining a community ACL" on page 1066.

Matching based on destination network

You can use the results of an IP ACL or an IP prefix list as the match condition.

To construct a route map that matches based on destination network, enter commands such as the following.

```
NetIron(config)# route-map NetMap permit 1
NetIron(config-routemap NetMap)# match ip address 1
```

Syntax: [**no**] **match ip address** <*ACL-name-or-num*>

Syntax: [**no**] **match ip address prefix-list** <*name*>

The <*name-or-num*> parameter with the first command specifies an IP ACL and can be a number from 1 – 199 or the ACL name if it is a named ACL. To configure an IP ACL, use the **ip access-list** or **access-list** command. Refer to the Access Control List chapter.

The <*name*> parameter with the second command specifies an IP prefix list name. To configure an IP prefix list, refer to "Defining and applying IP prefix lists" on page 1067.

Matching based on next-hop device

You can use the results of an IP ACL or an IP prefix list as the match condition.

To construct a route map that matches based on the next-hop device, enter commands such as the following.

```
NetIron(config)# route-map HopMap permit 1
NetIron(config-routemap HopMap)# match ip next-hop 2
```

Syntax: [**no**] **match ip next-hop** <*num*>

Syntax: [**no**] **match ip next-hop prefix-list** <*name*>

The <*num*> parameter with the first command specifies an IP ACL and can be a number from 1 – 199 or the ACL name if it is a named ACL. To configure an IP ACL, use the **ip access-list** or **access-list** command. Refer to the Access Control List chapter.

The <*name*> parameter with the second command specifies an IP prefix list name. To configure an IP prefix list, refer to "Defining and applying IP prefix lists" on page 1067.

Matching based on the route source

To match a BGP4 route based on its source, use the **match ip route-source** command.

```
NetIron(config)# access-list 10 permit 192.168.6.0 0.0.0.255
NetIron(config)# route-map bgp1 permit 1
NetIron(config-routemap bgp1)# match ip route-source 10
```

The first command configures an IP ACL that matches on routes received from 192.168.6.0/24. The remaining commands configure a route map that matches on all BGP4 routes advertised by the BGP4 neighbors whose addresses match addresses in the IP prefix list. You can add a set clause to change a route attribute in the routes that match. You also can use the route map as input for other commands, such as the **neighbor** and **network** commands and some show commands.

Syntax: [**no**] **match ip route-source** <*acl*> | **prefix** <*name*>

The <*acl*> | prefix <*name*> parameters specify the name or ID of an IP ACL, or an IP prefix list.

Matching on routes containing a specific set of communities

The device can match routes based on the presence of a community name or number in a route. To match based on a set of communities, configure a community ACL that lists the communities, then compare routes against the ACL.

```
NetIron(config)# ip community-list standard std_1 permit 12:34 no-export
NetIron(config)# route-map bgp2 permit 1
NetIron(config-routemap bgp2)# match community std_1 exact-match
```

The first command configures a community ACL that contains community number 12:34 and community name no-export. The remaining commands configure a route map that matches the community attributes field in BGP4 routes against the set of communities in the ACL. A route matches the route map only if the route contains all the communities in the ACL and no other communities.

Syntax: [**no**] **match community** <*acl*> **exact-match**

The <*acl*> parameter specifies the name of a community list ACL. You can specify up to five ACLs. Separate the ACL names or IDs with spaces.

```
NetIron(config)# ip community-list standard std_2 permit 23:45 56:78
NetIron(config)# route-map bgp3 permit 1
NetIron(config-routemap bgp3)# match community std_1 std_2 exact-match
```

These commands configure an additional community ACL, std_2, that contains community numbers 23:45 and 57:68. Route map bgp3 compares each BGP4 route against the sets of communities in ACLs std_1 and std_2. A BGP4 route that contains ***either but not both*** sets of communities matches the route map. For example, a route containing communities 23:45 and 57:68 matches. However, a route containing communities 23:45, 57:68 and 12:34, or communities 23:45, 57:68, 12:34, and no-export does not match. To match, the route communities must be the same as those in exactly one of the community ACLs used by the match community statement.

Matching based on BGP4 static network

The **match** option has been added to the **route-map** command that allows you to match on a BGP4 static network. In the following example, the route-map is configured to match on the BGP4 static network. The device is then configured to advertise to the core BGP4 peer (IP address 192.168.6.0) only the BGP4 static routes and nothing else.

```
NetIron(config)# route-map policygroup3 permit 10
NetIron(config-routemap policygroup3)# match protocol bgp static-network
NetIron(config-routemap policygroup3)# set localpref 150
NetIron(config-routemap policygroup3)# community no-export
NetIron(config-routemap policygroup3)# exit
NetIron(config)# router bgp
NetIron(config-bgp)# neighbor 192.168.6.0 route-map out policymap3
```

Syntax: [**no**] **match protocol bgp** [**external**|**internal**|**static-network**]

The **match protocol bgp external** option will match the eBGP routes.

The **match protocol bgp internal** option will match the iBGP routes.

The **match protocol bgp static-network** option will match the static-network BGP4 route, applicable at BGP4 outbound policy only.

Matching based on interface

The *match* option has been added to the **route-map** command that distributes any routes that have their next hop out one of the interfaces specified. This feature operates with the following conditions:

- The **match interface** option can only use the interface name (for example ethernet 1/2) and not the IP address as an argument.
- The **match interface** option is only effective during redistribution and does not apply for other route map usage such as: bgp outbound route update policy.
- The **match interface** option can be applied to other types of redistribution such as redistributing OSPF routes to BGP4, or filtering out all OSPF routes that point to a specific interface.

To configure the match-interface option, use the following command.

```
NetIron(config)# route-map test-route permit 99
NetIron(config-routemap test-route)# match interface ethernet 1/1 pos 3/2
NetIron(config-routemap test-route)# exit
```

Syntax: [**no**] **match interface** <*interface*> <*interface*>...

The <*interface*> variable specifies the interface that you want to use with the **match interface** command. Up to 5 interfaces of the following types can be specified:

- **ethernet** <*slot/port*>
- **loopback** <*loopback-number*>
- **null0**
- **pos** <*slot/port*>
- **tunnel** <*tunnel-ID*>
- **ve** <*ve-ID*>

Setting parameters in the routes

Use the following command to define a set clause that prepends an AS number to the AS path on each route that matches the corresponding match statement.

```
NetIron(config-routemap GET_ONE)# set as-path prepend 65535
```

Syntax: [**no**] **set**
[**as-path** [**prepend** <*as-num,as-num,...*>]] |
[**automatic-tag**] |
[**comm-list** <*acl*> **delete**] |
[**community** <*num*>:<*num*> | <*num*> | **internet** | **local-as** | **no-advertise** | **no-export**] |
[**dampening** [<*half-life*> <*reuse*> <*suppress*> <*max-suppress-time*>]]
[**ip next hop** <*ip-addr*>]
[**ip next-hop peer-address**] |
[**local-preference** <*num*>] |
[**metric** [**+** | **-**]<*num*> | **none**] |
[**metric-type type-1** | **type-2**] | **external**
[**metric-type internal**] |
[**next-hop** <*ip-addr*>] |
[origin igp | **incomplete**] |
[**tag**] |
[**weight** <*num*>]

The **as-path prepend** <*num,num,...*> parameter adds the specified AS numbers to the front of the AS-path list for the route. The range of <num> values is 1 – 65535 for two-byte ASNs and 1 – 4294967295 if AS4s have been enabled.

The **automatic-tag** parameter calculates and sets an automatic tag value for the route.

NOTE
This parameter applies only to routes redistributed into OSPF.

The **comm-list** parameter deletes a community from the community attributes field for a BGP4 route.

The **community** parameter sets the community attribute for the route to the number or well-known type you specify.

The **dampening** [<*half-life*> <*reuse*> <*suppress*> <*max-suppress-time*>] parameter sets route dampening parameters for the route. The <*half-life*> parameter specifies the number of minutes after which the route penalty becomes half its value. The <*reuse*> parameter specifies how low a route penalty must become before the route becomes eligible for use again after being suppressed. The <*suppress*> parameter specifies how high a route penalty can become before the device suppresses the route. The <*max-suppress-time*> parameter specifies the maximum number of minutes that a route can be suppressed regardless of how unstable it is. For information and examples, refer to "Configuring route flap dampening" on page 1028.

The **ip next hop** <*ip-addr*> parameter sets the next-hop IP address for route that matches a match statement in the route map.

The **ip next-hop peer-address** parameter sets the BGP4 next hop for a route to the neighbor address.

The **local-preference** <*num*> parameter sets the local preference for the route. You can set the preference to a value from 0 – 4294967295.

The **metric** [**+** | **-**] <*num*> | **none** parameter sets the MED (metric) value for the route. The default MED value is 0. You can set the preference to a value from 0 – 4294967295.

- **set metric** <*num*> – Sets the metric for the route to the number you specify.
- **set metric +** <*num*> – Increases route metric by the number you specify.
- **set metric -** <*num*> – Decreases route metric by the number you specify.
- **set metric none** – Removes the metric from the route (removes the MED attribute from the BGP4 route).

The **metric-type type-1** | **type-2** parameter changes the metric type of a route redistributed into OSPF.

The **metric-type internal** parameter sets the route MED to the same value as the IGP metric of the BGP4 next-hop route. The parameter does this when advertising a BGP4 route to an EBGP neighbor.

The **next-hop** <*ip-addr*> parameter sets the IP address of the route next-hop device.

The **origin igp incomplete** parameter sets the route's origin to IGP or INCOMPLETE.

The **tag** parameter is a keyword that sets the tag to be an AS-path attribute.

NOTE
This parameter applies only to routes redistributed into OSPF.

NOTE
You also can set the tag value using a table map. The table map changes the value only when the device places the route in the IP route table instead of changing the value in the BGP4 route table. Refer to "Using a table map to set the tag value" on page 1057.

The **weight** <*num*> parameter sets the weight for the route. The range for the weight value is 0 – 4294967295.

Setting a BGP4 route MED to equal the next-hop route IGP metric

To set a route's MED to the same value as the IGP metric of the BGP4 next-hop route, when advertising the route to a neighbor, enter commands such as the following.

```
NetIron(config)# access-list 1 permit 192.168.9.0 0.0.0.255
NetIron(config)# route-map bgp4 permit 1
NetIron(config-routemap bgp4)# match ip address 1
NetIron(config-routemap bgp4)# set metric-type internal
```

The first command configures an ACL that matches on routes with destination network 192.168.9.0. The remaining commands configure a route map that matches on the destination network in ACL 1, then sets the metric type for those routes to the same value as the IGP metric of the BGP4 next-hop route.

Syntax: [**no**] **set metric-type internal**

Setting the next-hop of a BGP4 route

To set the next-hop address of a BGP4 route to a neighbor address, enter commands such as the following.

```
NetIron(config)# route-map bgp5 permit 1
NetIron(config-routemap bgp5)# match ip address 1
NetIron(config-routemap bgp5)# set ip next-hop peer-address
```

These commands configure a route map that matches on routes whose destination network is specified in ACL 1, and sets the next hop in the routes to the neighbor address (inbound filtering) or the local IP address of the BGP4 session (outbound filtering).

Syntax: [no] **set ip next-hop peer-address**

The value that the software substitutes for **peer-address** depends on whether the route map is used for inbound filtering or outbound filtering:

- When you use the **set ip next-hop peer-address** command in an inbound route map filter, **peer-address** substitutes for the neighbor IP address.
- When you use the **set ip next-hop peer-address** command in an outbound route map filter, **peer-address** substitutes for the local IP address of the BGP4 session.

NOTE
You can use this command for a peer group configuration.

Deleting a community from a BGP4 route

To delete a community from a BGP4 route's community attributes field, enter commands such as the following.

```
NetIron(config)# ip community-list standard std_3 permit 12:99 12:86
NetIron(config)# route-map bgp6 permit 1
NetIron(config-routemap bgp6)# match ip address 1
NetIron(config-routemap bgp6)# set comm-list std_3 delete
```

The first command configures a community ACL containing community numbers 12:99 and 12:86. The remaining commands configure a route map that matches on routes whose destination network is specified in ACL 1, and deletes communities 12:99 and 12:86 from those routes. The route does not need to contain all the specified communities in order for them to be deleted. For example, if a route contains communities 12:86, 33:44, and 66:77, community 12:86 is deleted.

Syntax: [no] **set comm-list** <*acl*> **delete**

The <*acl*> parameter specifies the name of a community list ACL.

Route-map continue clauses for BGP4 routes

A continuation clause in a route-map directs program flow to skip over route-map instances to another, user-specified instance. If a matched instance contains a continue clause, the system looks for the instance that is identified in the continue clause.

The continue clause in a matching instance initiates another traversal at the instance that you specify in the continue clause. The system records all of the matched instances and, if no deny statements are encountered, proceeds to execute the set clauses of the matched instances.

If the system scans all route map instances but finds no matches, or if a deny condition is encountered, then it does not update the routes. Whenever a matched instance contains a deny parameter, the current traversal terminates, and none of the updates specified in the set clauses of the matched instances in both current and previous traversals are applied to the routes.

This feature supports a more programmable route map configuration and route filtering scheme for BGP4 peering. It can also execute additional instances in a route map after an instance is executed with successful match clauses. You can configure and organize more modular policy definitions to reduce the number of instances that are repeated within the same route map.

This feature currently applies to BGP4 routes only. For protocols other than BGP4, continue statements are ignored.

Specifying route-map continuation clauses

This section describes the configuration of route-map continuation clauses. The following sequence of steps (with referenced items in the screen output in bold) is described:

- The configuration context for a route-map named *test* is entered.
- Two route-map **continue** statements are added to route-map *test*.
- The **show route-map** output displays the modified route-map *test*.
- Subsequent **neighbor** commands identify the route map *test* in the inbound and outbound directions for the neighbor at 8.8.8.3.
- The **show ip bgp config** output shows inbound and outbound route-map *test* for the neighbor at 8.8.8.3.

```
NetIron(config-bgp)# route-map test permit 1
NetIron(config-routemap test)# match metric 10
NetIron(config-routemap test)# set weight 10
NetIron(config-routemap test)# continue 2
NetIron(config-routemap test)# route-map test permit 2
NetIron(config-routemap test)# match tag 10
NetIron(config-routemap test)# set weight 20
NetIron(config-routemap test)# continue 3
NetIron(config-routemap test)# router bgp
NetIron(config-bgp)# exit
NetIron(config-bgp)# show route-map test
route-map test permit 1
 match metric 10
 set weight 10
 continue 2
route-map test permit 2
 match tag 10
 set weight 20
 continue 3
NetIron(config-bgp)# neighbor 8.8.8.3 route-map in test
NetIron(config-bgp)# neighbor 8.8.8.3 route-map out test
NetIron(config-bgp)# show ip bgp config
Current BGP configuration:
router bgp
 local-as 100
 neighbor 8.8.8.3 remote-as 200
 address-family ipv4 unicast
 neighbor 8.8.8.3 route-map in test
 neighbor 8.8.8.3 route-map out test
 exit-address-family
 address-family ipv4 multicast
 exit-address-family
 address-family ipv6 unicast
 exit-address-family
 address-family ipv6 multicast
 exit-address-family
 address-family vpnv4 unicast
 exit-address-family
end of BGP configuration
```

Syntax: [no] **route-map** <*map-name*> **permit** | **deny** <*num*>

The **no** form of the command deletes the route map. The <*map-name*> is a string of up to 32 characters that specifies the map.

The **permit** option means the device applies match and set clauses associated with this route map instance.

The **deny** option means that any match causes the device to ignore the route map.

The <*num*> parameter specifies the instance of the route map defined in the route-map context that the CLI enters. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

Syntax: [no] **continue** [<*instance-number*>]

The **continue** command is entered in the context of a route-map instance. The [**no**] form of the command deletes the continue clause specified by <*instance-number*>. The instance number range is 0 – 4294967295, and the occurrences of <*instance-number*> must be in ascending numeric order. If you specify a continue clause without an instance number, it means "continue to the next route-map instance."

Syntax: [**no**] **neighbor** <*ip-addr*> | <*peer-group-name*> [**route-map in** | **out** <*map-name*>]

This syntax shows only the **neighbor** parameters that apply to this example. The <*ip-addr*> or <*peer-group-name*> identifies the neighbor, and the [**route-map in** | **out** <*map-name*>] option lets you specify a route map and direction to apply to the neighbor.

Dynamic route filter update

Routing protocols use various route filters to control the distribution of routes. Route filters are used to filter routes received from and advertised to other devices. Protocols also use route-map policies to control route redistribution from other routing protocols. In addition, route filter policies are used to select routes to be installed in the routing tables, and used by forwarding engine to forward traffic.

There are currently 6 different types of route filters defined for use in a device:

- Access List (ACL)
- Prefix-List
- BGP4 as-path Access-list
- BGP4 community-list
- BGP4 extended community-list
- Route-map

Not every protocol uses all of these route filters. A protocol will usually use two or three filter types. The filters used by BGP4, OSPF, RIP, IS-IS, RIPng, OSPFv3, MSDP, and MCast protocols are described in Table 168.

TABLE 168 Route filters used by each protocol

Protocol	Route map	Prefix list	Community- list	Extended community- list	As-path access- list	ACL
BGP4	X	X	BGP4 does not use Community- List filters directly. It does use them indirectly through route-map filters that contain Community-List filters.	BGP4 does not directly use Extended Community-List filters, but indirectly uses them through route-map filters that contain Extended Community-List filters.	X	X
OSPF	X	X				X
RIP	X	X				
IS-IS	X	X				
RIPng		X				
OSPFv3	X	X				
MSDP	X					
MCast						X

When a route filter is changed (created, modified or deleted) by a user, the filter change notification will be sent to all relevant protocols, so that protocols can take appropriate actions. For example if BGP4 is using a route-map (say MapX) to control the routes advertised to a particular peer, the change of route-map (MapX) will cause BGP4 to re-evaluate the advertised routes, and make the appropriate advertisements or withdrawals according to the new route-map policy.

A route filter change action can happen in three ways.

1. A new filter is defined (created).

   This filter name may be already referenced by an application. The application needs to be notified of the addition of the new filter, and will bind to and use the new filter. In general, if a filter name is referenced by an application, but is not actually defined, the application assumes the default **deny** action for the filter.

2. An existing filter is undefined (removed).

   If the deleted filter is already used and referenced by an application, the application will unbind itself from the deleted filter.

3. An existing filter is modified (updated).

   If the filter is already used and referenced by an application, the application will be notified.

Protocols are automatically notified when a route filter is created, deleted or modified. In addition, when a protocol is notified of a filter change, appropriate steps are taken to apply the new or updated filter to existing routes.

Commands for dynamic route filter updating

In order to allow multiple filter updates to be processed together by applications, the device waits 10 seconds by default before notifying applications of the filter change. You can force an immediate update notification or modify the time delay from when a change is made to a route filter to when the protocols are notified.

Route filter update delay settings can be configured using the commands shown here.

Setting a time delay for route filter update notification

Set the amount of time that the device waits before sending filter additions and modification notification to protocols using the following command.

```
NetIron# filter-change-update-delay 100
```

Syntax: [**no**] **filter-change-update-delay** <*delay-time*>

The <*delay-time*> variable specifies the amount of time in seconds that the device waits before sending filter addition and modification notification to protocols. The valid range is 0 to 600 seconds. If you set the value to 0, filter change notifications will not be automatically sent to protocols. The default value is 10 seconds.

NOTE
The **filter-change-update-delay** command also affects a route map that is being used in a PBR policy.

NOTE
Filter deletion notifications are sent to protocols without a delay.

Performing an immediate route filter update

To force an immediate filter update to the relevant protocols, use the following command.

```
NetIron# clear filter-change-update
```

Syntax: clear filter-change-update

This command forces an immediate filter update regardless of the filter-change-update-delay setting. It can also be used to simultaneously submit multiple change notifcations when the filter-change-update-delay is set to 0. When changes are complete, run the **clear filter-change-update** command to update protocols.

NOTE
There may be delays in sending route filter change notifications to applications, and delays in applying the new or updated filter to all existing routes retroactively. However any *new* routes or changes to existing routes will be subject to the new filters.

Filter update delay and BGP

The filter-changes-update-delay command applies *only* to *changes* of filters that are already used or referenced by applications. If the content of a filter is changed, the new filter action takes effect after **filter-changes-update-delay** for *existing* routes. The notification delay does *not* apply to situations where the usage or reference of a filter is changed in a protocol.

For example, the following BGP neighbor command sets or changes the route-map filter on a neighbor:

```
NetIron(config-bgp)# neighbor x.x.x.x route-map <map_abc> out
```

In this case, the router applies the route-map “map_abc” to the peer immediately, and automatically applies the new route-map filter retroactively to existing routes.

NOTE
The auto-update action for a BGP peer filter is newly introduced in release 3.5. In previous releases, a user needs to manually issue the **clear ip bgp neighbor x.x.x.x soft out** command to cause the router to apply the new route-map retroactively to existing routes.

The general guideline is to define a policy *first*, then apply it to a BGP peer.

BGP4 policy processing order

The order of application of policies when processing inbound and outbound route advertisements on the device is:

1. Ip prefix-list
2. Outbound Ip prefix-list ORF, if negotiated
3. Outbound extended-community ORF, if negotiated
4. Filter-list (using As-path access-list)
5. Distribute list (using IP ACL - ipv4 unicast only)
6. Route-map

Configuring cooperative BGP4 route filtering

By default, the device performs all filtering of incoming routes locally, on the device itself. You can use cooperative BGP4 route filtering to cause the filtering to be performed by a neighbor before it sends the routes to the device. Cooperative filtering conserves resources by eliminating unnecessary route updates and filter processing. For example, the device can send a deny filter to a neighbor, which the neighbor uses to filter out updates before sending them to the device. The neighbor saves the resources it would otherwise use to generate the route updates, and the device saves the resources it would use to filter out the routes.

When you enable cooperative filtering, the device advertises this capability in its Open message to the neighbor when initiating the neighbor session. The Open message also indicates whether the device is configured to send filters, receive filters, or both, and the types of filters it can send or receive. The device sends the filters as Outbound Route Filters (ORFs) in route refresh messages.

To configure cooperative filtering, perform the following tasks on the device and on the BGP4 neighbor:

- Configure the filter.

  NOTE
  Cooperative filtering is currently supported only for filters configured using IP prefix lists.

- Apply the filter as an ***inbound*** filter to the neighbor.
- Enable the cooperative route filtering feature on the device. You can enable the device to send ORFs to the neighbor, to receive ORFs from the neighbor, or both. The neighbor uses the ORFs you send as outbound filters when it sends routes to the device. Likewise, the device uses the ORFs it receives from the neighbor as outbound filters when sending routes to the neighbor.
- Reset the BGP4 neighbor session to send and receive ORFs.
- Perform these steps on the other device.

NOTE
If the device has inbound filters, the filters are still processed even if equivalent filters have been sent as ORFs to the neighbor.

Enabling cooperative filtering

To configure cooperative filtering, enter commands such as the following.

```
NetIron(config)# ip prefix-list Routesfrom1234 deny 20.20.0.0/24
NetIron(config)# ip prefix-list Routesfrom1234 permit 0.0.0.0/0 le 32
NetIron(config)# router bgp
NetIron(config-bgp)# neighbor 1.2.3.4 prefix-list Routesfrom1234 in
NetIron(config-bgp)# neighbor 1.2.3.4 capability orf prefixlist send
```

The first two commands configure statements for the IP prefix list Routesfrom1234. The first command configures a statement that denies routes to 20.20.20./24. The second command configures a statement that permits all other routes. Once you configure an IP prefix list statement, all routes not explicitly permitted by statements in the prefix list are denied.

The next two commands change the CLI to the BGP4 configuration level, then apply the IP prefix list to neighbor 1.2.3.4. The last command enables the device to send the IP prefix list as an ORF to neighbor 1.2.3.4. When the device sends the IP prefix list to the neighbor, the neighbor filters out the 20.20.0.x routes from its updates to the device. This assumes that the neighbor is also configured for cooperative filtering.

Syntax: [**no**] **neighbor** <*ip-addr*> | <*peer-group-name*> **capability orf prefixlist** [**send** | **receive**]

The <*ip-addr*> | <*peer-group-name*> parameters specify the IP address of a neighbor or the name of a peer group of neighbors.

The **send** | **receive** parameters specify the support you are enabling:

- **send** – The device sends the IP prefix lists to the neighbor.
- **receive** – The device accepts filters from the neighbor.

If you do not specify the capability, both capabilities are enabled.

The **prefixlist** parameter specifies the type of filter you want to send to the neighbor.

NOTE
The current release supports cooperative filtering only for filters configured using IP prefix lists.

Sending and receiving ORFs

Cooperative filtering affects neighbor sessions that start after the filtering is enabled, but do not affect sessions that are already established.

To activate cooperative filtering, reset the session with the neighbor. This is required because the cooperative filtering information is exchanged in Open messages during the start of a session.

To place a prefix-list change into effect after activating cooperative filtering, perform a soft reset of the neighbor session. A soft reset does not end the current session, but sends the prefix list to the neighbor in the next route refresh message.

NOTE
Make sure cooperative filtering is enabled on the device and on the neighbor before you send the filters.

To reset a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
NetIron# clear ip bgp neighbor 1.2.3.4
```

This command resets the BGP4 session with neighbor 1.2.3.4 and sends the ORFs to the neighbor. If the neighbor sends ORFs to the device, the device accepts them if the send capability is enabled.

To perform a soft reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
NetIron# clear ip bgp neighbor 1.2.3.4 soft in prefix-list
```

Syntax: **clear ip bgp neighbor** <*ip-addr*> [**soft in prefix-filter**]

If you use the **soft in prefix-filter** parameter, the device sends the updated IP prefix list to the neighbor as part of its route refresh message to the neighbor.

NOTE
If the device or the neighbor is not configured for cooperative filtering, the command sends a normal route refresh message.

Displaying cooperative filtering information

You can display the following cooperative filtering information:

- The cooperative filtering configuration on the device.
- The ORFs received from neighbors.

To display the cooperative filtering configuration on the device, enter a command such as the following. The line shown in bold type shows the cooperative filtering status.

```
NetIron# show ip bgp neighbor 10.10.10.1
1   IP Address: 10.10.10.1, AS: 65200 (IBGP), RouterID: 10.10.10.1
    State: ESTABLISHED, Time: 0h0m7s, KeepAliveTime: 60, HoldTime: 180
       RefreshCapability: Received
       CooperativeFilteringCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       0       1         0            1
       Received: 1       0       1         0            1
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
       Byte Sent:   110, Received: 110
       Local host:  10.10.10.2, Local  Port: 8138
       Remote host: 10.10.10.1, Remote Port: 179
       ISentSeq:        460  SendNext:        571  TotUnAck:          0
       TotSent:         111  ReTrans:           0  UnAckSeq:        571
       IRcvSeq:        7349  RcvNext:        7460  SendWnd:       16384
       TotalRcv:        111  DupliRcv:          0  RcvWnd:        16384
       SendQue:           0  RcvQue:            0  CngstWnd:       5325
```

Syntax: show ip bgp neighbor <*ip-addr*>

To display the ORFs received from a neighbor, enter a command such as the following:

```
NetIron# show ip bgp neighbor 10.10.10.1 received prefix-filter
ip prefix-list 10.10.10.1: 4 entries
     seq 5 permit 10.10.0.0/16 ge 18 le 28
     seq 10 permit 20.20.10.0/24
     seq 15 permit 30.0.0.0/8 le 32
     seq 20 permit 40.10.0.0/16 ge 18
```

Syntax: show ip bgp neighbor <*ip-addr*> **received prefix-filter**

Configuring route flap dampening

A route flap is a change in the state of a route, from up to down or down to up. A route state change causes changes in the route tables of the devices that support the route. Frequent route state changes can cause Internet instability and add processing overhead to the devices that support the route.

Route flap dampening helps reduce the impact of route flap by changing the way a BGP4 device responds to route state changes. When route flap dampening is configured, the device suppresses unstable routes until the number of route state changes drops enough to meet an acceptable degree of stability. The Dell implementation of route flap dampening is based on RFC 2439.

Route flap dampening is disabled by default. You can enable the feature globally or on an individual route basis using route maps.

NOTE
The device applies route flap dampening only to routes learned from EBGP neighbors.

The route flap dampening mechanism is based on penalties. When a route exceeds a configured penalty value, the device stops using that route and stops advertising it to other devices. The mechanism also allows route penalties to reduce over time if route stability improves.

The route flap dampening mechanism uses the following parameters:

- **Suppression threshold** – Specifies the penalty value at which the device stops using the route. Each time a route becomes unreachable or is withdrawn by a BGP4 UPDATE from a neighbor, the route receives a penalty of 1000. By default, when a route penalty is greater than 2000, the device stops using the route. By default, if a route goes down more than twice, the device stops using the route. You can set the suppression threshold to a value from 1 – 20000. The default is 2000.
- **Half-life** – Once a route has been assigned a penalty, the penalty decreases exponentially and decreases by half after the half-life period. The default half-life period is 15 minutes. The software reduces route penalties every five seconds. For example, if a route has a penalty of 2000 and does not receive any more penalties during the half-life, the penalty is reduced to 1000 after the half-life expires. You can configure the half-life to be from 1 – 45 minutes. The default is 15 minutes.
- **Reuse threshold** – Specifies the minimum penalty a route can have and still be suppressed by the device. If the route penalty falls below this value, the device un-suppresses the route and can use it again. The software evaluates the dampened routes every ten seconds and un-suppresses the routes that have penalties below the reuse threshold. You can set the reuse threshold to a value from 1 – 20000. The default is 750.
- **Maximum suppression time** – Specifies the maximum number of minutes a route can be suppressed regardless of how unstable the route has been before this time. You can set the parameter to a value from 1 – 20000 minutes. The default is four times the half-life. When the half-life value is set to its default (15 minutes), the maximum suppression time defaults to 60 minutes.

You can configure route flap dampening globally or for individual routes using route maps. If you configure route flap dampening parameters globally and also use route maps, the settings in the route maps override the global values.

Using a route map to configure route flap dampening for specific routes

Route maps enable you to fine tune route flap dampening parameters for individual routes. To configure route flap dampening parameters using route maps, configure BGP4 address filters for each route you want to set the dampening parameters for, then configure route map entries that set the dampening parameters for those routes. The following sections show examples.

Using a route map to configure route flap dampening for a specific neighbor

You can use a route map to configure route flap dampening for a specific neighbor by performing the following tasks:

- Configure an empty route map with no match or set clauses. This route map does not specify particular routes for dampening but does allow you to enable dampening globally when you refer to this route map from within the BGP4 configuration level.
- Configure another route map that explicitly enables dampening. Use a set clause within the route map to enable dampening. When you associate this route map with a specific neighbor, the route map enables dampening for all routes associated with the neighbor. You also can use match clauses within the route map to selectively perform dampening on some routes from the neighbor.

NOTE
You still need to configure the first route map to enable dampening globally. The second route map does not enable dampening by itself; it just applies dampening to a neighbor.

- Apply the route map to the neighbor.

To enable route flap dampening for a specific BGP4 neighbor, enter commands such as the following.

```
NetIron(config)# route-map DAMPENING_MAP_ENABLE permit 1
NetIron(config-routemap DAMPENING_MAP_ENABLE)# exit
NetIron(config)# route-map DAMPENING_MAP_NEIGHBOR_A permit 1
NetIron(config-routemap DAMPENING_MAP_NEIGHBOR_A)# set dampening
NetIron(config-routemap DAMPENING_MAP_NEIGHBOR_A)# exit
NetIron(config)# router bgp
NetIron(config-bgp)# dampening route-map DAMPENING_MAP_ENABLE
NetIron(config-bgp)# neighbor 10.10.10.1 route-map in DAMPENING_MAP_NEIGHBOR_A
```

In this example, the first command globally enables route flap dampening. This route map does not contain any match or set clauses. At the BGP4 configuration level, the **dampening route-map** command refers to the DAMPENING_MAP_ENABLE route map created by the first command, thus enabling dampening globally.

The third and fourth commands configure a second route map that explicitly enables dampening. Notice that the route map does not contain a match clause. The route map implicitly applies to all routes. Since the route map will be applied to a neighbor at the BGP4 configuration level, the route map will apply to all routes associated with the neighbor.

Although the second route map enables dampening, the first route map is still required. The second route map enables dampening for the neighbors to which the route map is applied. However, unless dampening is already enabled globally by the first route map, the second route map has no effect.

The last two commands apply the route maps. The **dampening route-map** command applies the first route map, which enables dampening globally. The **neighbor** command applies the second route map to neighbor 10.10.10.1. Since the second route map does not contain match clauses for specific routes, the route map enables dampening for all routes received from the neighbor.

Removing route damping from a route

You can un-suppress routes by removing route flap damping from the routes. The device allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# clear ip bgp damping
```

Syntax: clear ip bgp damping [<*ip-addr*> <*ip-mask*>]

The <*ip-addr*> parameter specifies a particular network.

The <*ip-mask*> parameter specifies the network's mask.

To un-suppress a specific route, enter a command such as the following.

```
NetIron# clear ip bgp damping 209.157.22.0 255.255.255.0
```

This command un-suppresses only the routes for network 209.157.22.0/24.

Displaying and clearing route flap dampening statistics

The software provides many options for displaying and clearing route flap statistics.

Displaying route flap dampening statistics

To display route dampening statistics or all the dampened routes, enter the following command at any CLI level.

```
NetIron# show ip bgp flap-statistics
Total number of flapping routes: 414
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flaps Since    Reuse    Path
h>  192.50.206.0/23    166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  203.255.192.0/20   166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  203.252.165.0/24   166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  192.50.208.0/23    166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  133.33.0.0/16      166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
*>  204.17.220.0/24    166.90.213.77   1     0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ip bgp flap-statistics [**regular-expression** <*regular-expression*> | <*address*> <*mask*> [**longer-prefixes**] | **neighbor** <*ip-addr*>]

The **regular-expression** <*regular-expression*> parameter is a regular expression. Regular expressions are the same ones supported for BGP4 AS-path filters. Refer to "Using regular expressions" on page 1064.

The <*address*> <*mask*> parameters specify a particular route. If you also use the optional **longer-prefixes** parameter, all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, all routes with the prefix 209.157. or longer (such as 209.157.22.) are displayed.

The **neighbor** <*ip-addr*> parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbor** <*ip-addr*> **flap-statistics**.

This display shows the following information.

TABLE 169 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the BGP4 route table that have changed state and have been marked as flapping routes.
Status code	Indicates the dampening status of the route, which can be one of the following: • > – This is the best route among those in the BGP4 route table to the route destination. • d – This route is currently dampened, and unusable. • h – The route has a history of flapping and is unreachable now. • * – The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The neighbor that sent the route to the device.
Flaps	The number of flaps the route has experienced.
Since	The amount of time since the first flap of this route.
Reuse	The amount of time remaining until this route will be un-suppressed and can be used again.
Path	Shows the AS-path information for the route.

You also can display all dampened routes by entering the **show ip bgp dampened-paths** command.

Clearing route flap dampening statistics

Clearing the dampening statistics for a route does not change the dampening status of the route. To clear all the route dampening statistics, enter the following command at any level of the CLI.

```
NetIron# clear ip bgp flap-statistics
```

Syntax: **clear ip bgp flap-statistics** [**regular-expression** <*regular-expression*> | <*address*> <*mask*> | **neighbor** <*ip-addr*>]

The parameters are the same as those for the **show ip bgp flap-statistics** command (except the **longer-prefixes** option is not supported). Refer to "Configuring route flap dampening" on page 1028.

NOTE
The **clear ip bgp damping** command not only clears statistics but also un-suppresses the routes. Refer to "Configuring route flap dampening" on page 1028.

Generating traps for BGP4

You can enable and disable SNMP traps for BGP4. BGP4 traps are enabled by default.

To enable BGP4 traps after they have been disabled, enter the following command.

```
NetIron(config)# snmp-server enable traps bgp
```

Syntax: [**no**] **snmp-server enable traps bgp**

Use the **no** form of the command to disable BGP4 traps.

Updating route information and resetting a neighbor session

The following sections describe how to update route information with a neighbor, reset a session with a neighbor, and close a session with a neighbor.

Any change to a policy (ACL, route map, and so on) is automatically applied to outbound routes that are learned from a BGP4 neighbor or peer group after the policy change occurs. However, you must reset the neighbor to update existing outbound routes.

Any change to a policy is automatically applied to inbound routes that are learned after the policy change occurs. However, to apply the changes to existing inbound routes (those inbound routes that were learned before the policy change), you must reset the neighbors to update the routes using one of the following methods:

- Request the complete BGP4 route table from the neighbor or peer group. You can use this method if the neighbor supports the refresh capability (RFCs 2842 and 2858). Most devices today support this capability.
- Clear (reset) the session with the neighbor or peer group. This is the only method you can use if soft reconfiguration is enabled for the neighbor.

You also can clear and reset the BGP4 routes that have been installed in the IP route table. Refer to "Clearing and resetting BGP4 routes in the IP route table" on page 1096.

Using soft reconfiguration

The ***soft reconfiguration*** feature applies policy changes without resetting the BGP4 session. Soft reconfiguration does not request the neighbor or group to send the entire BGP4 table, nor does the feature reset the session with the neighbor or group. Instead, soft reconfiguration stores all the route updates received from the neighbor or group. When you request a soft reset of inbound routes, the software performs route selection by comparing the policies against the stored route updates, instead of requesting the neighbor BGP4 route table or resetting the session with the neighbor.

When you enable the soft reconfiguration feature, it sends a refresh message to the neighbor or group if the neighbor or group supports dynamic refresh. Otherwise, the feature resets the neighbor session. This step is required to ensure that the soft reconfiguration feature has a complete set of updates to use, and occurs only once, when you enable the feature. The feature accumulates all the route updates from the neighbor, eliminating the need for additional refreshes or resets when you change policies in the future.

To use soft reconfiguration:

- Enable the feature.
- Make the policy changes.
- Apply the changes by requesting a soft reset of the inbound updates from the neighbor or group.

Enabling soft reconfiguration

To configure a neighbor for soft reconfiguration, enter a command such as the following.

```
NetIron(config-bgp)# neighbor 10.10.200.102 soft-reconfiguration inbound
```

This command enables soft reconfiguration for updates received from 10.10.200.102. The software dynamically resets the session with the neighbor, then retains all route updates from the neighbor following the reset.

Syntax: [**no**] **neighbor** <*ip-addr*> | <*peer-group-name*> **soft-reconfiguration inbound**

NOTE
The syntax related to soft reconfiguration is shown. For complete command syntax, refer to "Configuring BGP4 neighbors" on page 1037 and "Configuring a BGP4 peer group" on page 1046.

Placing a policy change into effect

To place policy changes into effect, enter a command such as the following.

```
NetIron(config-bgp)# clear ip bgp neighbor 10.10.200.102 soft in
```

This command updates the routes by comparing the route policies against the route updates that the device has stored. The command does not request additional updates from the neighbor or otherwise affect the session with the neighbor.

Syntax: **clear ip bgp neighbor** <*ip-addr*> | <*peer-group-name*> **soft in**

NOTE
If you do not specify **in**, the command applies to both inbound and outbound updates.

NOTE
The syntax related to soft reconfiguration is shown. For complete command syntax, refer to "Dynamically refreshing routes" on page 1093.

Displaying the filtered routes received from the neighbor or peer group

When you enable soft reconfiguration, the device saves all updates received from the specified neighbor or peer group, including updates that contain routes that are filtered out by the BGP4 route policies in effect on the device. To display the routes that have been filtered out, enter the following command at any level of the CLI.

```
NetIron# show ip bgp filtered-routes
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 1
3      4.60.212.0/22      192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 701 1 189
```

The routes displayed are the routes that were filtered out by the BGP4 policies on the device. The device did not place the routes in the BGP4 route table, but did keep the updates. If a policy change causes these routes to be permitted, the device does not need to request the route information from the neighbor, but instead uses the information in the updates.

Syntax: **show ip bgp filtered-routes** [<*ip-addr*>] | [**as-path-access-list** <*num*>] | [**detail**] | [**prefix-list** <*string*>]

The <*ip-addr*> parameter specifies the IP address of the destination network.

The **as-path-access-list** <*num*> parameter specifies an AS-path ACL. Only the routes permitted by the AS-path ACL are displayed.

The **detail** parameter displays detailed information for the routes. (The example shows summary information.) You can specify any of the other options after **detail** to further refine the display request.

The **prefix-list** <*string*> parameter specifies an IP prefix list. Only routes permitted by the prefix list are displayed.

NOTE
The syntax for displaying filtered routes is shown. For complete command syntax, refer to "Displaying the BGP4 route table" on page 1121.

Displaying all the routes received from the neighbor

To display all the route information received in route updates from a neighbor since you enabled soft reconfiguration, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp neighbor 192.168.4.106 routes
       There are 97345 received routes from  neighbor 192.168.4.106
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 1
3      4.60.212.0/22      192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8          192.168.4.106              100        0      BE
```

Syntax: show ip bgp neighbors <*ip-addr*> **received-routes** [*detail*]

The **detail** parameter displays detailed information for the routes. This example shows summary information.

NOTE
The syntax for displaying received routes is shown. For complete command syntax, refer to "Displaying BGP4 neighbor information" on page 1109.

Dynamically requesting a route refresh from a BGP4 neighbor

You can easily apply changes to filters that control BGP4 routes received from or advertised to a neighbor, without resetting the BGP4 session between the device and the neighbor. For example, if you add, change, or remove a BGP4 IP prefix list that denies specific routes received from a neighbor, you can apply the filter change by requesting a route refresh from the neighbor. If the neighbor also supports dynamic route refreshes, the neighbor resends its Adj-RIB-Out, its table of BGP4 routes. Using the route refresh feature, you do not need to reset the session with the neighbor.

The route refresh feature is based on the following specifications:

- RFC 2842. This RFC specifies the Capability Advertisement, which a BGP4 device uses to dynamically negotiate a capability with a neighbor.
- RFC 2858 for Multi-protocol Extension.
- RFC 2918, which describes the dynamic route refresh capability

The dynamic route refresh capability is enabled by default and cannot be disabled. When the device sends a BGP4 OPEN message to a neighbor, the device includes a Capability Advertisement to inform the neighbor that the device supports dynamic route refresh.

NOTE
The option for dynamically refreshing routes received from a neighbor requires the neighbor to support dynamic route refresh. If the neighbor does not support this feature, the option does not take effect and the software displays an error message. The option for dynamically re-advertising routes to a neighbor does not require the neighbor to support dynamic route refresh.

Dynamically refreshing routes

The following sections describe how to refresh BGP4 routes dynamically to put new or changed filters into effect.

To request a dynamic refresh of all routes from a neighbor, enter a command such as the following.

```
NetIron(config-bgp)# clear ip bgp neighbor 192.168.1.170 soft in
```

This command asks the neighbor to send its BGP4 table (Adj-RIB-Out) again. The device applies its filters to the incoming routes and adds, modifies, or removes BGP4 routes as necessary.

Syntax: **clear ip bgp neighbor all** | <*ip-addr*> | <*peer-group-name*> | <*as-num*> [**soft-outbound** | **soft** [**in** | **out**]]

The **all** | <*ip-addr*> | <*peer-group-name*> | <*as-num*> parameters specify the neighbor. The <*ip-addr*> parameter specifies a neighbor by its IP interface with the device. The <*peer-group-name*> specifies all neighbors in a specific peer group. The <*as-num*> parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor.

The **soft** [**in** | **out**] parameter specifies whether you want to refresh the routes received from the neighbor or sent to the neighbor:

- **soft in** does one of the following:
  - If you enabled soft reconfiguration for the neighbor or peer group, **soft in** updates the routes by comparing the route policies against the route updates that the device has stored. Soft reconfiguration does not request additional updates from the neighbor or otherwise affect the session with the neighbor. Refer to "Using soft reconfiguration" on page 1090.
  - If you did not enable soft reconfiguration, **soft in** requests the entire BGP4 route table for the neighbor (Adj-RIB-Out), then applies the filters to add, change, or exclude routes.
  - If a neighbor does not support dynamic refresh, **soft in** resets the neighbor session.
- **soft out** updates all outbound routes, then sends the entire BGP4 router table for the device (Adj-RIB-Out) to the neighbor, after changing or excluding the routes affected by the filters.

If you do not specify **in** or **out**, the device performs both options.

NOTE
The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor. The **soft out** parameter updates all outbound routes, then sends the entire BGP4 route table for the device (Adj-RIB-Out) to the neighbor, after changing or excluding the routes affected by the filters. Use **soft-outbound** if only the outbound policy is changed.

To dynamically resend all the device BGP4 routes to a neighbor, enter a command such as the following.

```
NetIron(config-bgp)# clear ip bgp neighbor 192.168.1.170 soft out
```

This command applies filters for outgoing routes to the device BGP4 route table (Adj-RIB-Out), changes or excludes routes accordingly, then sends the resulting Adj-RIB-Out to the neighbor.

NOTE
The device does not automatically update outbound routes using a new or changed outbound policy or filter when a session with the neighbor goes up or down. Instead, the device applies a new or changed policy or filter when a route is placed in the outbound queue (Adj-RIB-Out).

To place a new or changed outbound policy or filter into effect, you must enter a **clear ip bgp neighbor** command regardless of whether the neighbor session is up or down. You can enter the command without optional parameters or with the **soft out** or **soft-outbound** option. Either way, you must specify a parameter for the neighbor (<*ip-addr*>, <*as-num*>, <*peer-group-name*>, or **all**).

Displaying dynamic refresh information

You can use the **show ip bgp neighbors** command to display information for dynamic refresh requests. For each neighbor, the display lists the number of dynamic refresh requests the device has sent to or received from the neighbor and indicates whether the device received confirmation from the neighbor that the neighbor supports dynamic route refresh.

The RefreshCapability field indicates whether this device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability. The statistics in the Message Sent and Message Received rows under Refresh-Req indicate how many dynamic refreshes have been sent to and received from the neighbor. The statistic is cumulative across sessions.

```
NetIron(config-bgp)# show ip bgp neighbor 10.4.0.2
1   IP Address: 10.4.0.2, AS: 5 (EBGP), RouterID: 100.0.0.1
      Description: neighbor 10.4.0.2
    State: ESTABLISHED, Time: 0h1m0s, KeepAliveTime: 0, HoldTime: 0
      PeerGroup: pg1
      Mutihop-EBGP: yes, ttl: 1
      RouteReflectorClient: yes
      SendCommunity: yes
      NextHopSelf: yes
      DefaultOriginate: yes (default sent)
      MaximumPrefixLimit: 90000
      RemovePrivateAs: : yes
      RefreshCapability: Received
    Route Filter Policies:
      Distribute-list: (out) 20
      Filter-list: (in) 30
      Prefix-list: (in) pf1
      Route-map: (in) setnp1  (out) setnp2
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
      Sent    : 1       1       1         0            0
      Received: 1       8       1         0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                 Tx: 0h0m59s    ---           Rx: 0h0m59s    ---
    Last Connection Reset Reason:Unknown
      Notification Sent:     Unspecified
      Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
      Byte Sent:   115, Received: 492
      Local host:  10.4.0.1, Local  Port: 179
      Remote host: 10.4.0.2, Remote Port: 8053
      ISentSeq:   52837276  SendNext:   52837392  TotUnAck:          0
      TotSent:         116  ReTrans:           0  UnAckSeq:   52837392
      IRcvSeq:  2155052043  RcvNext:  2155052536  SendWnd:       16384
      TotalRcv:        493  DupliRcv:          0  RcvWnd:        16384
      SendQue:           0  RcvQue:            0  CngstWnd:       1460
```

Closing or resetting a neighbor session

You can close a neighbor session or resend route updates to a neighbor.

If you make changes to filters or route maps and the neighbor does not support dynamic route refresh, use the following methods to ensure that neighbors contain only the routes you want them to contain:

- If you close a neighbor session, the device and the neighbor clear all the routes they learned from each other. When the device and neighbor establish a new BGP4 session, they exchange route tables again. Use this method if you want the device to relearn routes from the neighbor and resend its own route table to the neighbor.
- If you use the soft-outbound option, the device compiles a list of all the routes it would normally send to the neighbor at the beginning of a session. However, before sending the updates, the device also applies the filters and route maps you have configured to the list of routes. If the filters or route maps result in changes to the list of routes, the device sends updates to advertise, change, or even withdraw routes on the neighbor as needed. This ensures that the neighbor receives only the routes you want it to contain. Even if the neighbor already contains a route learned from the device that you later decided to filter out, using the soft-outbound option removes that route from the neighbor.

You can specify a single neighbor or a peer group.

To close a neighbor session and thus flush all the routes exchanged by the device and the neighbor, enter the following command.

```
NetIron# clear ip bgp neighbor all
```

Syntax: clear ip bgp neighbor all | <*ip-addr*> | <*peer-group-name*> | <*as-num*> [**soft-outbound** | **soft** [**in** | **out**]]

The **all** | <*ip-addr*> | <*peer-group-name*> | <*as-num*> parameters specify the neighbor. The <*ip-addr*> parameter specifies a neighbor by its IP interface with the device. The <*peer-group-name*> specifies all neighbors in a specific peer group. The <*as-num*> parameter specifies all neighbors within an AS and has a range of 1 – 4294967295. The **all** keyword specifies all neighbors.

To resend routes to a neighbor without closing the neighbor session, enter a command such as the following.

```
NetIron# clear ip bgp neighbor 10.0.0.1 soft out
```

Clearing and resetting BGP4 routes in the IP route table

To clear BGP4 routes from the IP route table and reset the routes, enter a command such as the following.

```
NetIron# clear ip bgp routes
```

Syntax: clear ip bgp routes [<*ip-addr*>/<*prefix-length*>]

Clearing traffic counters

You can clear the counters (reset them to 0) for BGP4 messages.

To clear the BGP4 message counter for all neighbors, enter the following command.

```
NetIron# clear ip bgp traffic
```

Syntax: clear ip bgp traffic

To clear the BGP4 message counter for a specific neighbor, enter a command such as the following.

```
NetIron# clear ip bgp neighbor 10.0.0.1 traffic
```

To clear the BGP4 message counter for all neighbors within a peer group, enter a command such as the following.

```
NetIron# clear ip bgp neighbor PeerGroup1 traffic
```

Syntax: clear ip bgp neighbor all | <*ip-addr*> | <*peer-group-name*> | <*as-num*> **traffic**

The **all** | <*ip-addr*> | <*peer-group-name*> | <*as-num*> parameters specify the neighbor. The <*ip-addr*> parameter specifies a neighbor by its IP interface with the device. The <*peer-group-name*> specifies all neighbors in a specific peer group. The <*as-num*> parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

Clearing route flap dampening statistics

Clearing the dampening statistics for a route does not change the dampening status of the route.

To clear all the route dampening statistics, enter the following command at any level of the CLI.

```
NetIron# clear ip bgp flap-statistics
```

Syntax: **clear ip bgp flap-statistics** [**regular-expression** <*regular-expression*> | <*address*> <*mask*> | **neighbor** <*ip-addr*>]

The parameters are the same as those for the **show ip bgp flap-statistics** command (except the **longer-prefixes** option is not supported). Refer to "Displaying route flap dampening statistics" on page 1129.

NOTE
The **clear ip bgp damping** command not only clears statistics but also un-suppresses the routes. Refer to "Displaying route flap dampening statistics" on page 1129.

Removing route flap dampening

You can un-suppress routes by removing route flap dampening from the routes. The device allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# clear ip bgp damping
```

Syntax: **clear ip bgp damping** [<*ip-addr*> <*ip-mask*>]

The <*ip-addr*> parameter specifies a particular network.

The <*ip-mask*> parameter specifies the network's mask.

To un-suppress a specific route, enter a command such as the following:

```
NetIron# clear ip bgp damping 209.157.22.0 255.255.255.0
```

This command un-suppresses only the routes for network 209.157.22.0/24.

Clearing diagnostic buffers

The device stores the following BGP4 diagnostic information in buffers:

- The first 400 bytes of the last packet received that contained an error
- The last NOTIFICATION message either sent or received by the device

To display these buffers, use options with the **show ip bgp neighbors** command. Refer to "Displaying BGP4 neighbor information" on page 1109.

This information can be useful if you are working with Dell Technical Support to resolve a problem. The buffers do not identify the system time when the data was written to the buffer. If you want to ensure that diagnostic data in a buffer is recent, you can clear the buffers. You can clear the buffers for a specific neighbor or for all neighbors.

If you clear the buffer containing the first 400 bytes of the last packet that contained errors, all the bytes are changed to zeros. The Last Connection Reset Reason field of the BGP4 neighbor table also is cleared.

If you clear the buffer containing the last NOTIFICATION message sent or received, the buffer contains no data.

You can clear the buffers for all neighbors, for an individual neighbor, or for all the neighbors within a specific peer group.

To clear these buffers for neighbor 10.0.0.1, enter the following commands.

```
NetIron# clear ip bgp neighbor 10.0.0.1 last-packet-with-error
NetIron# clear ip bgp neighbor 10.0.0.1 notification-errors
```

Syntax: **clear ip bgp neighbor all** | <*ip-addr*> | <*peer-group-name*> | <*as-num*> **last-packet-with-error** | **notification-errors**

The **all** | <*ip-addr*> | <*peer-group-name*> | <*as-num*> parameters specify the neighbor. The <*ip-addr*> parameter specifies a neighbor by its IP interface with the device. The <*peer-group-name*> specifies all neighbors in a specific peer group. The <*as-num*> parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

Configuring BGP4 Restart

BGP4 Restart can be configured for a global routing instance or for a specified Virtual Routing and Forwarding (VRF) instance. The following sections describe how to enable the BGP4 Restart feature.

Configuring BGP4 restart for the global routing instance

Use the following command to enable the BGP4 restart feature globally on a device.

```
NetIron(config)# router bgp
NetIron(config-bgp)# graceful-restart
```

Syntax: [**no**] **graceful-restart**

Configuring BGP4 Restart for a VRF

Use the following command to enable the BGP4 restart feature for a specified VRF.

```
NetIron(config)# router bgp
NetIron(config-bgp)# address-family ipv4 unicast vrf blue
NetIron(config-bgp-ipv4u-vrf)# graceful-restart
```

Syntax: [**no**] **graceful-restart**

Configuring timers for BGP4 Restart (optional)

You can optionally configure the following timers to change their values from the default values:

- Restart Timer
- Stale Routes Timer
- Purge Timer

The <*seconds*> variable sets the maximum restart wait time advertised to neighbors. Possible values are 1– 3600 seconds. The default value is 120 seconds.

Configuring the restart timer for BGP4 Restart

Use the following command to specify the maximum amount of time a device will maintain routes from and forward traffic to a restarting device.

```
NetIron(config-bgp)# graceful-restart restart-timer 150
```

Syntax: [no] **graceful-restart restart-timer** <*seconds*>

The <*seconds*> variable sets the maximum restart wait time advertised to neighbors. Possible values are 1 - 3600 seconds. The default value is 120 seconds.

Configuring BGP4 Restart stale routes timer

Use the following command to specify the maximum amount of time a helper device will wait for an end-of-RIB message from a peer before deleting routes from that peer.

```
NetIron(config-bgp)# graceful-restart stale-routes-time 120
```

Syntax: [no] **graceful-restart stale-routes-time** <*seconds*>

The <*seconds*> variable sets the maximum time before a helper device cleans up stale routes. Possible values are 1 - 3600 seconds. The default value is 360 seconds.

Configuring BGP4 Restart purge timer

Use the following command to specify the maximum amount of time a device will maintain stale routes in its routing table before purging them.

```
NetIron(config-bgp)# graceful-restart purge-time 900
```

Syntax: [no] **graceful-restart purge-time** <*seconds*>

The <*seconds*> variable sets the maximum time before a restarting device cleans up stale routes. Possible values are 1 – 3600 seconds. The default value is 600 seconds.

For information about displaying BGP4 restart neighbor information, refer to "Displaying BGP4 restart neighbor information" on page 1131.

Configuring BGP4 null0 routing

BGP4 null0 routing is described in "BGP4 null0 routing" on page 1007. The following example configures a null0 routing application to stop denial of service attacks from remote hosts on the Internet.

Configuration steps

1. Select a device, for example, device 6, to distribute null0 routes throughout the BGP4 network.
2. Configure a route-map to match a particular tag (50) and set the next-hop address to an unused network address (192.168.0.1).
3. Set the local-preference to a value higher than any possible internal or external local-preference (50).

4. Complete the route map by setting origin to IGP.
5. On device 6, redistribute the static routes into BGP4, using route-map <*route-map-name*> (redistribute static route-map block user).
6. On device 1, (the device facing the Internet), configure a null0 route matching the next-hop address in the route-map (ip route 192.168.0.1/32 null0).
7. Repeat step 3 for all devices interfacing with the Internet (edge corporate devices). In this case, device 2 has the same null0 route as device 1.
8. On device 6, configure the network prefixes associated with the traffic you want to drop. The static route IP address references a destination address. You must point the static route to the egress port, (for example, Ethernet 3/7), and specify the tag 50, matching the route-map configuration.

Configuration examples

Device 6

The following configuration defines specific prefixes to filter:

```
NetIron(config)# ip route 110.0.0.40/29 ethernet 3/7 tag 50
NetIron(config)# ip route 115.0.0.192/27 ethernet 3/7 tag 50
NetIron(config)# ip route 120.014.0/23 ethernet 3/7 tag 50
```

The following configuration redistributes routes into BGP4.

```
NetIron(config)# router bgp
NetIron(config-bgp-router)# local-as 100
NetIron(config-bgp-router)# neighbor <router1_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router2_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router3_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router4_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router5_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router7_int_ip address> remote-as 100
NetIron(config-bgp-router)# redistribute static route-map blockuser
NetIron(config-bgp-router)# exit
```

The following configuration defines the specific next hop address and sets the local preference to preferred.

```
NetIron(config)# route-map blockuser permit 10
NetIron(config-routemap blockuser)# match tag 50
NetIron(config-routemap blockuser)# set ip next-hop 192.168.0.1
NetIron(config-routemap blockuser)# set local-preference 1000000
NetIron(config-routemap blockuser)# set origin igp
NetIron(config-routemap blockuser)# exit
```

NOTE
A match tag can take up to 16 tags. During the execution of a route-map, a match on any tag value in the list is considered a successful match.

Device 1

The following configuration defines the null0 route to the specific next hop address. The next hop address 192.168.0.1 points to 128.178.1.101, which gets blocked.

```
NetIron(config)# ip route 192.168.0.1/32 null0
```

```
NetIron(config)# router bgp
local-as 100
NetIron(config-bgp-router)# neighbor <router2_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router3_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router4_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router5_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router6_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router7_int_ip address> remote-as 100
```

Device 2

The following configuration defines a null0 route to the specific next hop address. The next hop address 192.168.0.1 points to 128.178.1.101, which gets blocked.

```
NetIron(config)# ip route 192.168.0.1/32 null0
NetIron(config)# router bgp
NetIron(config-bgp-router)# local-as 100
NetIron(config-bgp-router)# neighbor <router1_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router3_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router4_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router5_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router6_int_ip address> remote-as 100
NetIron(config-bgp-router)# neighbor <router7_int_ip address> remote-as 100
```

Show commands

After configuring the null0 application, you can display the output using **show** commands.

Device 6

Show ip route static output for device 6.

```
NetIron# show ip route static
Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
          Destination          Gateway         Port        Cost       Type
1       110.0.0.40/29          DIRECT          eth 3/7     1/1        S
2       115.0.0.192/27         DIRECT          eth 3/7     1/1        S
3       120.0.14.0/23          DIRECT          eth 3/7     1/1        S
NetIron#
```

Device 1 and 2

Show ip route static output for device 1 and device 2.

```
NetIron# show ip route static
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
             Destination        Gateway         Port        Cost       Type
1            192.168.0.1/32     DIRECT          drop        1/1        S
NetIron#
```

Device 6

Show BGP4 routing table output for Device-6

```
NetIron#show ip bgp route
Total number of BGP Routes: 126
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED  E:EBGP
H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED s: STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      30.0.1.0/24        40.0.1.3        0          100        0      BI
         AS_PATH:
.            ..                      .                      .          .
9      110.0.0.16/30      90.0.1.3                  100        0      I
         AS_PATH: 85
10     110.0.0.40/29      192.168.0.1     1        1000000 32768  BL
         AS_PATH:
11     110.0.0.80/28      90.0.1.3                  100        0      I
 .           ..                        .                      .          .
 .           ..                        .                    .          .
36     115.0.0.96/28      30.0.1.3                  100        0      I
         AS_PATH: 50
37     115.0.0.192/27     192.168.0.1     1       10000000 32768  BL
         AS_PATH:
 .            ..                     .                      .          .
64     120.0.7.0/24       70.0.1.3                  100        0      I
         AS_PATH: 10
65     120.0.14.0/23      192.168.0.1     1        1000000 32768  BL
         AS_PATH: ..                   .                      .          .
```

Device 1 and 2

The **show ip route** output for device 1 and device 2 shows “drop” under the Port column for the network prefixes you configured with null0 routing

```
NetIron#show ip route
Total number of IP routes: 133
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
        Destination        Gateway         Port        Cost       Type
1       9.0.1.24/32        DIRECT          loopback 1  0/0    D
2       30.0.1.0/24        DIRECT          eth 2/7     0/0        D
3       40.0.1.0/24        DIRECT          eth 2/1     0/0        D
.
13      110.0.0.6/31       90.0.1.3        eth 2/2     20/1       B
14      110.0.0.16/30      90.0.1.3        eth 2/2     20/1       B
15      110.0.0.40/29      DIRECT          drop        200/0      B
.          ..               .               .          .          .
42      115.0.0.192/27     DIRECT          drop        200/0      B
43      115.0.1.128/26     30.0.1.3        eth 2/7     20/1       B
.          ..               .               .          .          .
69      120.0.7.0/24       70.0.1.3        eth 2/10    20/1       B
70      120.0.14.0/23      DIRECT          drop        200/0      B
.          ..               .               .          .          .
.          ..               .               .          .          .
131     130.144.0.0/12     80.0.1.3        eth 3/4     20/1       B
132     192.168.0.1/32     DIRECT          drop        1/1        S
NetIron#
```

Generalized TTL Security Mechanism support

The device supports the Generalized TTL Security Mechanism (GTSM) as defined in RFC 3682. GTSM protects the device from attacks of invalid BGP4 control traffic that is sent to overload the CPU or hijack the BGP4 session. GTSM protection applies to EBGP neighbors only.

When GTSM protection is enabled, BGP4 control packets sent by the device to a neighbor have a Time To Live (TTL) value of 255. In addition, the device expects the BGP4 control packets received from the neighbor to have a TTL value of either 254 or 255. For multihop peers (where the **ebgp-multihop** option is configured for the neighbor), the device expects the TTL for BGP4 control packets received from the neighbor to be greater than or equal to 255, minus the configured number of hops to the neighbor. If the BGP4 control packets received from the neighbor do not have the anticipated value, the device drops them.

For more information on GTSM protection, see RFC 3682.

To enable GTSM protection for neighbor 192.168.9.210 (for example), enter the following command.

```
NetIron(config-bgp-router)# neighbor 192.168.9.210 ebgp-btsh
```

Syntax: [**no**] **neighbor** <*ip-addr*> | <*peer-group-name*> **ebgp-btsh**

NOTE
For GTSM protection to work properly, it must be enabled on both the device and the neighbor.

Displaying BGP4 information

You can display the following configuration information and statistics for BGP4 protocol:

- Summary BGP4 configuration information for the device
- Active BGP4 configuration information (the BGP4 information in the running configuration)
- Neighbor information
- Peer-group information
- Information about the paths from which BGP4 selects routes
- Summary BGP4 route information
- The device's BGP4 route table
- Route flap dampening statistics
- Active route maps (the route map configuration information in the running configuration)
- BGP4 Restart Neighbor Information
- AS4 support and asdot notation

Displaying summary BGP4 information

You can display the local AS number, the maximum number of routes and neighbors supported, and some BGP4 statistics. You can also display BGP4 memory usage for:

- BGP4 routes installed
- Routes advertising to all neighbors (aggregated into peer groups)
- Attribute entries installed

The **show ip bgp summary** command output has the following limitations:

- If a BGP4 peer is not configured for an address-family, the peer information is not displayed.
- If a BGP4 peer is configured for an address-family but not negotiated for an address-family after the BGP4 peer is in the established state, the **show ip bgp summary** command output shows (**NoNeg**) at the end of the line for this peer.
- If a BGP4 peer is configured and negotiated for that address-family, its display is the same as in previous releases.

To view summary BGP4 information for the device, enter the following command at any CLI prompt

```
NetIron# show ip bgp summary
  BGP4 Summary
  Router ID: 30.10.1.14 Local AS Number: 100
  Confederation Identifier: not configured
  Confederation Peers:
  Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
  Number of Neighbors Configured: 67, UP: 67
  Number of Routes Installed: 258088, Uses 22195568 bytes
  Number of Routes Advertising to All Neighbors: 17,035844 (3,099146 entries),
Uses 192,147052 bytes
  Number of Attribute Entries Installed: 612223, Uses 55100070 bytes
  Neighbor Address  AS#   State   Time     Rt:Accepted Filtered Sent   ToSend
  100.0.100.2       100   ESTABp  0h28m24s    0        0        258087 0
  100.0.101.2       100   ESTAB   0h28m24s    0        0        258087 0
  1.2.3.4           200   ADMDN   0h44m56s    0        0        0      2
```

Syntax: show ip bgp summary

This display shows the following information.

TABLE 170 BGP4 summary information

This field...	Displays...
Router ID	The device's device ID.
Local AS Number	The BGP4 AS number for the device.
Confederation Identifier	The AS number of the confederation in which the device resides.
Confederation Peers	The numbers of the local ASs contained in the confederation. This list matches the confederation peer list you configure on the device.
Maximum Number of Paths Supported for Load Sharing	The maximum number of route paths across which the device can balance traffic to the same destination. The feature is enabled by default but the default number of paths is 1. You can increase the number from 2 – 8 paths. Refer to "Configuring BGP4 multipath load sharing" on page 1034.
Number of Neighbors Configured	The number of BGP4 neighbors configured on this device, and currently in established state.
Number of Routes Installed	The number of BGP4 routes in the device BGP4 route table and the route or path memory usage.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors, the total number of unique ribout group entries, and the amount of memory used by these groups.
Number of Attribute Entries Installed	The number of BGP4 route-attribute entries in the device route-attributes table and the amount of memory used by these entries. To display the route-attribute table, refer to "Displaying BGP4 route-attribute entries" on page 1127.
Neighbor Address	The IP addresses of the BGP4 neighbors for this device.
AS#	The AS number.

TABLE 170 BGP4 summary information (Continued)

This field...	Displays...
State	The state of device sessions with each neighbor. The states are from this perspective of the device, not the neighbor. State values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each device: • IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. • ADMND – The neighbor has been administratively shut down. Refer to "Administratively shutting down a session with a BGP4 neighbor" on page 1049. • CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. **Note**: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT – BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM – BGP4 has received an Open message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the device receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED – BGP4 is ready to exchange UPDATE packets with the neighbor. **Operational States:** Additional information regarding the operational states of the BGP4 states described above may be added as described in the following: • **(+)** – is displayed if there is more BGP4 data in the TCP receiver queue. **Note**: If you display information for the neighbor using the **show ip bgp neighbor** <*ip-addr*> command, the TCP receiver queue value will be greater than 0. • **(-)** – indicates that the session has gone down and the software is clearing or removing routes. • **(*)** – indicates that the inbound or outbound policy is being updated for the peer. • **(s)** – indicates that the peer has negotiated restart, and the session is in a stale state. • **(r)** – indicates that the peer is restarting the BGP4 connection, through restart. • **(^)** – on the standby MP indicates that the peer is in the ESTABLISHED state and has received restart capability (in the primary MP). • **(<)** – indicates that the device is waiting to receive the "End of RIB" message the peer. • **(p)** – indicates that the neighbor ribout group membership change is pending or in progress
Time	The time that has passed since the state last changed.
Accepted	The number of routes received from the neighbor that this device installed in the BGP4 route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this device filtered out some of the routes received in the UPDATE messages.
Filtered	The routes or prefixes that have been filtered out: • If soft reconfiguration is enabled, this field shows how many routes were filtered out (not placed in the BGP4 route table) but retained in memory. • If soft reconfiguration is not enabled, this field shows the number of BGP4 routes that have been filtered out.

TABLE 170 BGP4 summary information (Continued)

This field...	Displays...
Sent	The number of BGP4 routes the device has sent to the neighbor.
ToSend	The number of routes the device has queued to advertise and withdraw to a neighbor.

Displaying the active BGP4 configuration

To view the active BGP4 configuration information contained in the running configuration without displaying the entire running configuration, enter the following command at any level of the CLI.

```
NetIron# show ip bgp config
router bgp
  local-as 200
neighbor 102.102.1.1 remote-as 200
neighbor 102.102.1.1 ebgp-multihop
neighbor 102.102.1.1 update-source loopback 1
neighbor 192.168.2.1 remote-as 100
neighbor 200.200.2.2 remote-as 400
neighbor 1000:2::1:1 remote-as 200
neighbor 2000:1::1:2 remote-as 400
neighbor 4444::1 remote-as 300

address-family ipv4 unicast
no neighbor 1000:2::1:1 activate
no neighbor 2000:1::1:2 activate
no neighbor 4444::1 activate
exit-address-family

address-family ipv4 multicast
exit-address-family

address-family ipv6 unicast
redistribute static
neighbor 1000:2::1:1 activate
neighbor 2000:1::1:2 activate
neighbor 4444::1 activate
exit-address-family
end of BGP configuration
```

Syntax: show ip bgp config

Displaying summary neighbor information

The **show ip bgp neighbor** command output has the following limitations.

1. If BGP4 peer is not configured for an address-family, the peer information will NOT be displayed.
2. If BGP4 peer is configured for an address-family, it will display the same as in previous releases.

To display summary neighbor information, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp)# show ip bgp neighbor 192.168.4.211 routes-summary
1   IP Address: 192.168.4.211
Routes Accepted/Installed:1,  Filtered/Kept:11,  Filtered:11
   Routes Selected as BEST Routes:1
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0

NLRIs Received in Update Message:24,  Withdraws:0 (0),  Replacements:1
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0

Routes Advertised:0,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:0,  Withdraws:0,  Replacements:0

Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0
```

Syntax: show ip bgp neighbors [<*ip-addr*>] | [**route-summary**]

This display shows the following information.

TABLE 171 BGP4 route summary information for a neighbor

This field...	Displays...
IP Address	The IP address of the neighbor.
Routes Received	How many routes the device has received from the neighbor during the current BGP4 session: • Accepted or Installed – Number of received routes the device accepted and installed in the BGP4 route table. • Filtered or Kept – Number of routes that were filtered out, but were retained in memory for use by the soft reconfiguration feature. • Filtered – Number of received routes filtered out.
Routes Selected as BEST Routes	The number of routes that the device selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next-hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages: • Withdraws – Number of withdrawn routes the device has received. • Replacements – Number of replacement routes the device has received.

TABLE 171 BGP4 route summary information for a neighbor (Continued)

This field...	Displays...
NLRIs Discarded due to	Indicates the number of times the device discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The configured maximum prefix amount had been reached. • AS Loop – An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • maxas-limit aspath – The number of route entries discarded because the AS path exceeded the configured maximum length or exceeded the internal memory limits. • Invalid Nexthop – The next-hop value was not acceptable. • Duplicated Originator_ID – The originator ID was the same as the local device ID. • Cluster_ID – The cluster list contained the local cluster ID, or the local device ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the device has advertised to this neighbor: • To be Sent – The number of routes queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the device has queued to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the device has sent to this neighbor in UPDATE messages: • Withdraws – Number of routes the device has sent to the neighbor to withdraw. • Replacements – Number of routes the device has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the device has run out of BGP4 memory for the neighbor during the current BGP4 session: • Receiving Update Messages – The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes (NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes – The number of times there was no memory for BGP4 attribute entries. • Outbound Routes (RIB-out) – The number of times there was no memory to place a “best” route into the neighbor route information base (Adj-RIB-Out) for routes to be advertised.

Displaying BGP4 neighbor information

You can display configuration information and statistics for BGP4 neighbors of the device.

To view BGP4 neighbor information, including the values for all the configured parameters, enter the following command.

NOTE
The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

```
NetIron(config-bgp)# show ip bgp neighbor 10.4.0.2
    Total number of BGP neighbors:
1   IP Address: 10.4.0.2, AS: 5 (EBGP), RouterID: 100.0.0.1
       Description: neighbor 10.4.0.2
    Local AS: 101
    State: ESTABLISHED, Time: 0h1m0s, KeepAliveTime: 0, HoldTime: 0
       PeerGroup: pg1
       Multihop-EBGP: yes, ttl: 1
       RouteReflectorClient: yes
       SendCommunity: yes
       NextHopSelf: yes
       DefaultOriginate: yes (default sent)
       MaximumPrefixLimit: 90000
       RemovePrivateAs: : yes
       RefreshCapability: Received
    Route Filter Policies:
       Distribute-list: (out) 20
       Filter-list: (in) 30
       Prefix-list: (in) pf1
       Route-map: (in) setnp1  (out) setnp2
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       1       1         0            0
       Received: 1       8       1         0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: 0h0m59s    ---           Rx: 0h0m59s    ---
    Last Connection Reset Reason:Unknown
      Notification Sent:     Unspecified
      Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
       Local host:  10.4.0.1, Local  Port: 179
       Remote host: 10.4.0.2, Remote Port: 8053
       ISentSeq:   52837276  SendNext:   52837392  TotUnAck:          0
       TotSent:         116  ReTrans:           0  UnAckSeq:   52837392
       IRcvSeq:  2155052043  RcvNext:  2155052536  SendWnd:       16384
       TotalRcv:        493  DupliRcv:          0  RcvWnd:        16384
       SendQue:           0  RcvQue:            0  CngstWnd:       1460
```

This example shows how to display information for a specific neighbor, by specifying the neighbor's IP address with the command. Since none of the other display options are used, all of the information is displayed for the neighbor. The number in the far left column indicates the neighbor for which information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the Transmission Control Block (TCB) for the TCP session between the device and the neighbor. These fields are described in detail in section 3.2 of RFC 793, "Transmission Control Protocol Functional Specification".

Syntax: **show ip bgp neighbors** [<*ip-addr*> [**advertised-routes** [**detail** [<*ip-addr*>[/<*mask-bits*>]]]] | [**attribute-entries** [**detail**]] | [**flap-statistics**] | [**last-packet-with-error**] | [**received prefix-filter**] |
[**received-routes**] | [**routes** [**best**] | [**detail** [**best**] | [**not-installed-best**] | [**unreachable**]] | [**rib-out-routes** [<*ip-addr*>/<*mask-bits*> | <*ip-addr*> <*net-mask*> | **detail**]] | [**routes-summary**]]

The <*ip-addr*> option lets you narrow the scope of the command to a specific neighbor.

The **advertised-routes** option displays only the routes that the device has advertised to the neighbor during the current BGP4 session.

The **attribute-entries** option shows the attribute-entries associated with routes received from the neighbor.

The **flap-statistics** option shows the route flap statistics for routes received from or sent to the neighbor.

The **last-packet-with-error** option displays the last packet from the neighbor that contained an error. The packet contents are displayed in decoded (human-readable) format.

The **received prefix-filter** option shows the Outbound Route Filters (ORFs) received from the neighbor. This option applies to cooperative route filtering.

The **received-routes** option lists all the route information received in route updates from the neighbor since the soft reconfiguration feature was enabled. Refer to "Using soft reconfiguration" on page 1090.

The **routes** option lists the routes received in UPDATE messages from the neighbor. You can specify the following additional options:

- **best** – Displays the routes received from the neighbor that the device selected as the best routes to their destinations.
- **not-installed-best** – Displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
- **unreachable** – Displays the routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
- **detail** – Displays detailed information for the specified routes. You can refine your information request by also specifying one of the options (**best**, **not-installed-best**, or **unreachable**).

The **rib-out-routes** option lists the route information base (RIB) for outbound routes. You can display all routes or specify a network address.

The **routes-summary** option displays a summary of the following information:

- Number of routes received from the neighbor
- Number of routes accepted by this device from the neighbor
- Number of routes this device filtered out of the UPDATES received from the neighbor and did not accept
- Number of routes advertised to the neighbor
- Number of attribute entries associated with routes received from or advertised to the neighbor.

This display shows the following information.

TABLE 172 BGP4 neighbor information

This field...	Displays...
Total Number of BGP4 Neighbors	The number of BGP4 neighbors configured.
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.

TABLE 172 BGP4 neighbor information (Continued)

This field...	Displays...
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP – The neighbor is in another AS. • EBGP_Confed – The neighbor is a member of another sub-AS in the same confederation. • IBGP – The neighbor is in the same AS.
RouterID	The neighbor device ID.
Description	The description you gave the neighbor when you configured it on the device.
Local AS	The value (if any) of the Local AS configured.
State	The state of the session with the neighbor. The states are from the device perspective, not the neighbor perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each device: • IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND – The neighbor has been administratively shut down. Refer to "Administratively shutting down a session with a BGP4 neighbor" on page 1049. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. **NOTE:** If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT – BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the device receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED – BGP4 is ready to exchange UPDATE messages with the neighbor. • If there is more BGP4 data in the TCP receiver queue, a plus sign (+) is also displayed. **NOTE:** If you display information for the neighbor using the **show ip bgp neighbor** <*ip-addr*> command, the TCP receiver queue value will be greater than 0.
Time	The amount of time this session has been in the current state.
KeepAliveTime	The keep alive time, which specifies how often this device sends keepalive messages to the neighbor. Refer to "Changing the Keep Alive Time and Hold Time" on page 1057.
HoldTime	The hold time, which specifies how many seconds the device will wait for a keepalive or update message from a BGP4 neighbor before deciding that the neighbor is not operational. Refer to "Changing the Keep Alive Time and Hold Time" on page 1057.
PeerGroup	The name of the peer group the neighbor is in, if applicable.

TABLE 172 BGP4 neighbor information (Continued)

This field...	Displays...
Multihop-EBGP	Whether this option is enabled for the neighbor.
RouteReflectorClient	Whether this option is enabled for the neighbor.
SendCommunity	Whether this option is enabled for the neighbor.
NextHopSelf	Whether this option is enabled for the neighbor.
DefaultOriginate	Whether this option is enabled for the neighbor.
MaximumPrefixLimit	Maximum number of prefixes the device will accept from this neighbor.
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
CooperativeFilteringCapability	Whether the neighbor is enabled for cooperative route filtering.
Distribute-list	Lists the distribute list parameters, if configured.
Filter-list	Lists the filter list parameters, if configured.
Prefix-list	Lists the prefix list parameters, if configured.
Route-map	Lists the route map parameters, if configured.
Messages Sent	The number of messages this device has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this device has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

TABLE 172 BGP4 neighbor information (Continued)

This field...	Displays...
Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following: Reasons described in the BGP4 specifications: • Message Header Error • Connection Not Synchronized • Bad Message Length • Bad Message Type • OPEN Message Error • Unsupported Version Number • Bad Peer AS Number • Bad BGP4 Identifier • Unsupported Optional Parameter • Authentication Failure • Unacceptable Hold Time • Unsupported Capability • UPDATE Message Error • Malformed Attribute List • Unrecognized Well-known Attribute • Missing Well-known Attribute • Attribute Flags Error • Attribute Length Error • Invalid ORIGIN Attribute • Invalid NEXT_HOP Attribute • Optional Attribute Error • Invalid Network Field • Malformed AS_PATH • Hold Timer Expired • Finite State Machine Error • Rcv Notification
Last Connection Reset Reason (cont.)	Reasons specific to the implementation: • Reset All Peer Sessions • User Reset Peer Session • Port State Down • Peer Removed • Peer Shutdown • Peer AS Number Change • Peer AS Confederation Change • TCP Connection KeepAlive Timeout • TCP Connection Closed by Remote • TCP Data Stream Error Detected

TABLE 172 BGP4 neighbor information (Continued)

This field...	Displays...

Notification Sent	If the device receives a notification message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.
	• Message Header Error:
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • Unspecified
	• Open Message Error:
	  • Unsupported Version
	  • Bad Peer As
	  • Bad BGP4 Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unspecified
	• Update Message Error:
	  • Malformed Attribute List
	  • Unrecognized Attribute
	  • Missing Attribute
	  • Attribute Flag Error
	  • Attribute Length Error
	  • Invalid Origin Attribute
	  • Invalid NextHop Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS Path
	  • Unspecified
	• Hold Timer Expired
	• Finite State Machine Error
	• Cease
	• Unspecified

Notification Received	See above.

TABLE 172 BGP4 neighbor information (Continued)

This field...	Displays...
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the device.
Local port	The TCP port the device is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the device.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the device that have not been acknowledged by the neighbor.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the device retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.

TABLE 172 BGP4 neighbor information (Continued)

This field...	Displays...
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying route information for a neighbor

You can display routes based on the following criteria:

- A summary of the routes for a specific neighbor.
- Routes received from the neighbor that the device selected as the best routes to their destinations.
- Routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
- Routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
- Routes for a specific network advertised by the device to the neighbor.
- The Routing Information Base (RIB) for a specific network advertised to the neighbor. You can display the RIB regardless of whether the device has already sent it to the neighbor.

Displaying summary route information

To display summary route information, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp)# show ip bgp neighbor 10.1.0.2 routes-summary
1   IP Address: 10.1.0.2
Routes Accepted/Installed:1,  Filtered/Kept:11,  Filtered:11
   Routes Selected as BEST Routes:1
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0

NLRIs Received in Update Message:24,  Withdraws:0 (0),  Replacements:1
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0

Routes Advertised:0,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:0,  Withdraws:0,  Replacements:0

Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0
```

This display shows the following information.

TABLE 173 BGP4 route summary information for a neighbor

This field...	Displays...
Routes Received	How many routes the device has received from the neighbor during the current BGP4 session: • Accepted or Installed – Indicates how many of the received routes the device accepted and installed in the BGP4 route table. • Filtered – Indicates how many of the received routes the device did not accept or install because they were denied by filters on the device.
Routes Selected as BEST Routes	The number of routes that the device selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages: • Withdraws – The number of withdrawn routes the device has received. • Replacements – The number of replacement routes the device has received.
NLRIs Discarded due to	Indicates the number of times the device discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The configured maximum prefix amount had been reached. • AS Loop – An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • Invalid Nexthop – The next-hop value was not acceptable. • Duplicated Originator_ID – The originator ID was the same as the local device ID. • Cluster_ID – The cluster list contained the local cluster ID, or contained the local device ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the device has advertised to this neighbor: • To be Sent – Number of routes the device has queued to send to this neighbor. • To be Withdrawn –Number of NLRIs for withdrawing routes the device has queued up to send to this neighbor in update messages.

TABLE 173 BGP4 route summary information for a neighbor (Continued)

This field...	Displays...
NLRIs Sent in Update Message	The number of NLRIs for new routes the device has sent to this neighbor in UPDATE messages: • Withdraws – The number of routes the device has sent to the neighbor to withdraw. • Replacements – The number of routes the device has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the device has run out of BGP4 memory for the neighbor during the current BGP4 session: • Receiving Update Messages – The number of times update messages were discarded because there was no memory for attribute entries. • Accepting Routes (NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes – The number of times there was no memory for BGP4 attribute entries. • Outbound Routes (RIB-out) – The number of times there was no memory to place a "best" route into the neighbor route information base (Adj-RIB-Out) for routes to be advertised.

Displaying advertised routes

To display the routes the device has advertised to a specific neighbor for a specific network, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp neighbors 192.168.4.211 advertised-routes
       There are 2 routes advertised to neighbor 192.168.4.211
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Network        Next Hop        Metric     LocPrf     Weight     Status
1      102.0.0.0/24   192.168.2.102   12                    32768      BL
2      200.1.1.0/24   192.168.2.102   0                     32768    BL
```

You also can enter a specific route.

```
NetIron# show ip bgp neighbors 192.168.4.211 advertised 200.1.1.0/24
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Network        Next Hop        Metric     LocPrf     Weight     Status
1      200.1.1.0/24   192.168.2.102   0                     32768      BL
```

Syntax: show ip bgp neighbor <*ip-addr*> **advertised-routes** [<*ip-addr*>/<*prefix*>]

For information about the fields in this display, refer to Table 173. The fields in this display also appear in the **show ip bgp** display.

Displaying the routes with destinations that are unreachable

To display BGP4 routes with destinations that are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI:

```
NetIron(config-bgp)# show ip bgp neighbor 192.168.4.211 routes unreachable
```

Syntax: show ip bgp neighbor <*ip-addr*> **routes unreachable**

For information about the fields in this display, refer to Table 173. The fields in this display also appear in the **show ip bgp** display.

Displaying the Adj-RIB-Out for a neighbor

To display the current BGP4 Routing Information Base (Adj-RIB-Out) for a specific neighbor and a specific destination network, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp)# show ip bgp neighbor 192.168.4.211 rib-out-routes
192.168.1.0/24
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      200.1.1.0/24       0.0.0.0         0          101        32768  BL
```

The Adj-RIB-Out contains the routes that the device either has most recently sent to the neighbor or is about to send to the neighbor.

Syntax: show ip bgp neighbor <*ip-addr*> **rib-out-routes** [<*ip-addr*>/<*prefix*>]

For information about the fields in this display, refer to Table 173. The fields in this display also appear in the **show ip bgp** display.

Displaying peer group information

To display peer-group information, enter a command such as the following at the Privileged EXEC level of the CLI.

```
NetIron# show ip bgp peer-group pg1
1   BGP peer-group is pg
    Description: peer group abc
       SendCommunity: yes
       NextHopSelf: yes
       DefaultOriginate: yes
    Members:
       IP Address: 192.168.10.10, AS: 65111
```

Syntax: show ip bgp peer-group [<*peer-group-name*>]

Only the parameters that have values different from their defaults are listed.

Displaying summary route information

To display summary statistics for all the routes in the device's BGP4 route table, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp)# show ip bgp routes summary
  Total number of BGP routes (NLRIs) Installed     : 20
  Distinct BGP destination networks                 : 20
  Filtered BGP routes for soft reconfig             : 100178
  Routes originated by this router                  : 2
  Routes selected as BEST routes                    : 19
  BEST routes not installed in IP forwarding table : 1
  Unreachable routes (no IGP route for NEXTHOP)     : 1
  IBGP routes selected as best routes               : 0
  EBGP routes selected as best routes               : 17
```

Syntax: show ip bgp routes summary

This display shows the following information.

TABLE 174 BGP4 summary route information

This field...	Displays...
Total number of BGP4 routes (NLRIs) Installed	Number of BGP4 routes the device has installed in the BGP4 route table.
Distinct BGP4 destination networks	Number of destination networks the installed routes represent. The BGP4 route table can have multiple routes to the same network.
Filtered BGP4 routes for soft reconfig	Number of route updates received from soft-reconfigured neighbors or peer groups that have been filtered out but retained. For information about soft reconfiguration, refer to "Using soft reconfiguration" on page 1090.
Routes originated by this device	Number of routes in the BGP4 route table that this device originated.
Routes selected as BEST routes	Number of routes in the BGP4 route table that this device has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	Number of BGP4 routes that are the best BGP4 routes to their destinations but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable routes (no IGP route for NEXTHOP)	Number of routes in the BGP4 route table whose destinations are unreachable because the next-hop is unreachable.
IBGP routes selected as best routes	Number of "best" routes in the BGP4 route table that are IBGP routes.
EBGP routes selected as best routes	Number of "best" routes in the BGP4 route table that are EBGP routes.

Displaying the BGP4 route table

BGP4 uses filters that you define as well as the algorithm described in "How BGP4 selects a path for a route" on page 1000 to determine the preferred route to a destination. BGP4 sends only the preferred route to the IP table. To view all the learned BGP4 routes, you can display the BGP4 table.

To view the BGP4 route table, enter the following command.

```
NetIron(config-bgp)# show ip bgp routes
Total number of BGP Routes: 97371
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 1
3      4.60.212.0/22      192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 3356 7170 1455
5      8.8.1.0/24         192.168.4.106   0           100        0      BE
         AS_PATH: 65001
```

Syntax: show ip bgp routes [[**network**] <*ip-addr*>] | <*num*> | [**age** <*secs*>] | [**as-path-access-list** <*num*>] | [**best**] | [**cidr-only**] | [**community** <*num*> | **no-export** | **no-advertise** | **internet** |

local-as] | [**community-access-list** <*num*>] | [**community-list** <*num*> | [**detail** <*option*>] | [**filter-list** <*num, num,...*>] |[**next-hop** <*ip-addr*>] | [**no-best**] | [**not-installed-best**] | [**prefix-list** <*string*>] | [**regular-expression** <*regular-expression*>] | [**route-map** <*map-name*>] | [**summary**] | [**unreachable**]

The <*ip-addr*> option displays routes for a specific network. The **network** keyword is optional. You can enter the network address without entering **network** in front of it.

The <*num*> option specifies the table entry with which you want the display to start. For example, if you want to list entries beginning with table entry 100, specify 100.

The **age** <*secs*> parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** <*num*> parameter filters the display using the specified AS-path ACL.

The **best** parameter displays the routes received from the neighbor that the device selected as the best routes to their destinations.

The **cidr-only** option lists only the routes whose network masks do not match their class network length.

The **community** option lets you display routes for a specific community. You can specify **local-as**, **no-export**, **no-advertise**, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of up to 1– 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** <*num*> parameter filters the display using the specified community ACL.

The **community-list** option lets you display routes that match a specific community filter.

The **detail** option lets you display more details about the routes. You can refine your request by also specifying one of the other display options after the detail keyword.

The **filter-list** option displays routes that match a specific address filter list.

The **next-hop** <*ip-addr*> option displays the routes for a given next-hop IP address.

The **no-best** option displays the routes for which none of the routes to a given prefix were selected as the best route. The IP route table does not contain a BGP4 route for any of the routes listed by the command.

The **not-installed-best** option displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).

The **prefix-list** <*string*> parameter filters the display using the specified IP prefix list.

The **regular-expression** <*regular-expression*> option filters the display based on a regular expression. Refer to "Using regular expressions" on page 1064.

The **route-map** <*map-name*> parameter filters the display by using the specified route map. The software displays only the routes that match the match clauses in the route map. Software disregards the route map's set clauses.

The **summary** option displays summary information for the routes.

The **unreachable** option displays the routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next-hop.

Displaying the best BGP4 routes

To display all the BGP4 routes in the device's BGP4 route table that are the best routes to their destinations, enter a command such as the following at any level of the CLI

```
NetIron(config-bgp)# show ip bgp routes best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 1
3      4.60.212.0/22      192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 3356 7170 1455
5      9.2.0.0/16         192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701
```

Syntax: show ip bgp routes best

For information about the fields in this display, refer to Table 173. The fields in this display also appear in the **show ip bgp** display.

Displaying BGP4 routes whose destinations are unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp)# show ip bgp routes unreachable
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix      Next Hop        Metric     LocPrf     Weight Status
1      8.8.8.0/24  192.168.5.1     0          101        0
         AS_PATH: 65001 4355 1
```

Syntax: show ip bgp routes unreachable

For information about the fields in this display, refer to Table 173. The fields in this display also appear in the **show ip bgp** display.

Displaying information for a specific route

To display BGP4 network information by specifying an IP address within the network, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp)# show ip bgp 9.3.4.0
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  9.3.4.0/24         192.168.4.106          100    0      65001 4355 1 1221 ?
       Last update to IP routing table: 0h11m38s, 1 path(s) installed:
         Gateway           Port
         192.168.2.1       2/1
       Route is advertised to 1 peers:
        20.20.20.2(65300)
```

Syntax: show ip bgp [**route**] <*ip-addr*>/<*prefix*> [**longer-prefixes**] | <*ip-addr*>

If you use the **route** option, the display for the information is different, as shown in the following example.

```
NetIron(config-bgp)# show ip bgp route 9.3.4.0
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      9.3.4.0/24         192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 1 1221
       Last update to IP routing table: 0h12m1s, 1 path(s) installed:
         Gateway           Port
         192.168.2.1       2/1
       Route is advertised to 1 peers:
        20.20.20.2(65300)
```

These displays show the following information.

TABLE 175 BGP4 network information

This field...	Displays...
Number of BGP4 Routes matching display condition	The number of routes that matched the display parameters you entered. This is the number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output. **NOTE:** This field appears only if you *do not* enter the **route** option.
Prefix	The network address and prefix.
Next Hop	The next-hop device for reaching the network.
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.

TABLE 175 BGP4 network information (Continued)

This field...	Displays...
Path	The route AS path. **NOTE:** This field appears only if you *do not* enter the **route** option.
Origin code	A character that indicates the route origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command output. **NOTE:** This field appears only if you *do not* enter the **route** option.
Status	The route status, which can be one or more of the following: • A – AGGREGATE.The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE:** If the "b" is lowercase, the software was not able to install the route in the IP route table. • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – INTERNAL. The route was learned through BGP4. • L – LOCAL. The route originated on this device. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is lowercase, the software was not able to install the route in the IP route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **NOTE:** This field appears only if you enter the **route** option.

Displaying route details

This example shows the information displayed when you use the **detail** option. In this example, the information for one route is shown.

```
NetIron# show ip bgp routes detail
Total number of BGP Routes: 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
1       Prefix: 10.5.0.0/24,  Status: BME,  Age: 0h28m28s
         NEXT_HOP: 201.1.1.2,  Learned from Peer: 10.1.0.2 (5)
          LOCAL_PREF: 101,  MED: 0,  ORIGIN: igp,  Weight: 10
           AS_PATH: 5
            Adj_RIB_out count: 4,  Admin distance 20
```

Syntax: show ip bgp routes detail

These displays show the following information.

TABLE 176 BGP4 route information

This field...	Displays...
Total number of BGP4 Routes	The number of BGP4 routes.
Status codes	A list of the characters that indicate route status. The status code is appears in the left column of the display, to the left of each route. The status codes are described in the command's output.
Prefix	The network prefix and mask length.
Status	The route status, which can be one or more of the following: • A – AGGREGATE.The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE:** If the "b" is lowercase, the software was not able to install the route in the IP route table. • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – INTERNAL. The route was learned through BGP4. • L – LOCAL. The route originated on this device. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is lowercase, the software was not able to install the route in the IP route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
Age	The last time an update occurred.
Next_Hop	The next-hop device for reaching the network.
Learned from Peer	The IP address of the neighbor that sent this route.
Local_Pref	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
MED	The route metric. If the route does not have a metric, this field is blank.

TABLE 176 BGP4 route information (Continued)

This field...	Displays...
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with these attributes came to BGP4 through EGP. • IGP – The routes with these attributes came to BGP4 through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	The value this device associates with routes from a specific neighbor. For example, if the device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Atomic	Whether network information in this route has been aggregated and this aggregation has resulted in information loss. **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Aggregation ID	The device that originated this aggregation.
Aggregation AS	The AS in which the network information was aggregated. This value applies only to aggregated routes and is otherwise 0.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this route has passed.
Learned From	The IP address of the neighbor from which the device learned the route.
Admin Distance	The administrative distance of the route.
Adj_RIB_out	The number of neighbors to which the route has been or will be advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4 neighbor.
Communities	The communities the route is in.

Displaying BGP4 route-attribute entries

The route-attribute entries table lists the sets of BGP4 attributes stored in device memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the device typically has fewer route attribute entries than routes.

To display the IP route table, enter the following command.

```
NetIron# show ip bgp attribute-entries
```

Syntax: show ip bgp attribute-entries

This example shows the information displayed by this command. A zero value indicates that the attribute is not set.

```
NetIron# show ip bgp attribute-entries
        Total number of BGP Attribute Entries: 7753
1       Next Hop  :192.168.11.1       Metric   :0                Origin:IGP
        Originator:0.0.0.0            Cluster List:None
         Aggregator:AS Number :0         Router-ID:0.0.0.0          Atomic:FALSE
        Local Pref:100                Communities:Internet
        AS Path   :(65002) 65001 4355 2548 3561 5400 6669 5548
2       Next Hop  :192.168.11.1       Metric   :0                Origin:IGP
        Originator:0.0.0.0            Cluster List:None
         Aggregator:AS Number :0         Router-ID:0.0.0.0          Atomic:FALSE
        Local Pref:100                Communities:Internet
        AS Path   :(65002) 65001 4355 2548
```

This display shows the following information.

TABLE 177 BGP4 route-attribute entries information

This field...	Displays...
Total number of BGP4 Attribute Entries	The number of routes contained in this BGP4 route table.
Next Hop	The IP address of the next-hop device for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with these attributes came to BGP4 through EGP. • IGP – The routes with these attributes came to BGP4 through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • AS Number shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • Router-ID shows the device that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated *and* this aggregation has resulted in information loss. • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use these attributes relative to other routes in the local AS.
Communities	The communities to which routes with these attributes belong.
AS Path	The ASs through which routes with these attributes have passed. The local AS is shown in parentheses.

Displaying the routes BGP4 has placed in the IP route table

The IP route table indicates the routes it has received from BGP4 by listing “BGP” as the route type.

To display the IP route table, enter the following command.

```
NetIron# show ip route
```

Syntax: show ip route [<*ip-addr*> | <*num*> | **bgp** | **ospf** | **rip** | **isis**]

This example shows the information displayed by this command. Notice that most of the routes in this example have type “B”, indicating that their source is BGP4.

```
NetIron# show ip route
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
       Destination         Gateway       Port     Cost    Type
1      130.130.130.0/24    11.11.11.1    ve 1     200/0   B
2      130.130.131.0/24    11.11.11.1     ve 1    200/0   B
```

Displaying route flap dampening statistics

To display route dampening statistics or all the dampened routes, enter the following command at any level of the CLI.

```
NetIron# show ip bgp flap-statistics
Total number of flapping routes: 414
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flaps Since    Reuse    Path
h>  192.50.206.0/23    166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  203.255.192.0/20   166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  203.252.165.0/24   166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  192.50.208.0/23    166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  133.33.0.0/16      166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
*>  204.17.220.0/24    166.90.213.77   1     0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ip bgp flap-statistics [**regular-expression** <*regular-expression*> | <*address*> <*mask*> [longer-prefixes] | **neighbor** <*ip-addr*> | **filter-list** <*num*>...]

The **regular-expression** <*regular-expression*> parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters. Refer to “Using regular expressions” on page 1064.

The <*address*> <*mask*> parameters specify a particular route. If you also use the optional **longer-prefixes** parameter, all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

The **neighbor** <*ip-addr*> parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You can also display route flap statistics for routes learned from a neighbor by entering the **show ip bgp neighbor** <*ip-addr*> **flap-statistics**.command.

The **filter-list** <*num*> parameter specifies one or more filters. Only routes that have been dampened and that match the specified filters are displayed.

This display shows the following information.

TABLE 178 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the BGP4 route table that have changed state and have been marked as flapping routes.
Status code	The dampening status of the route, which can be one of the following: • > – This is the best route among those in the BGP4 route table to the route destination. • d – This route is currently dampened, and thus unusable. • h – The route has a history of flapping and is unreachable now. • * – The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The neighbor that sent the route to this device.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time since the first flap of this route.
Reuse	The amount of time remaining until this route will be un-suppressed and thus be usable again.
Path	The AS-path information for the route.

You can display all dampened routes by entering the **show ip bgp dampened-paths**.command.

Displaying the active route map configuration

You can view the active route map configuration (contained in the running configuration) without displaying the entire running configuration.by entering the following command at any level of the CLI.

```
NetIron# show route-map
route-map permitnet4 permit 10
 match ip address prefix-list plist1
route-map permitnet1 permit 1
 match ip address prefix-list plist2
route-map setcomm permit 1
 set community 1234:2345 no-export
route-map test111 permit 111
 match address-filters 11
 set community 11:12 no-export
route-map permit1122 permit 12
 match ip address 11
route-map permit1122 permit 13
 match ip address std_22
```

This example shows that the running configuration contains six route maps. Notice that the match and set statements within each route map are listed beneath the command for the route map itself. In this simplified example, each route map contains only one match or set statement.

To display the active configuration for a specific route map, enter a command such as the following, which specifies a route map name.

```
NetIron# show route-map setcomm
route-map setcomm permit 1
 set community 1234:2345 no-export
```

This example shows the active configuration for a route map named “setcomm“.

Syntax: **show route-map** [<*map-name*>]

Displaying BGP4 restart neighbor information

To display BGP4 restart information for BGP4 neighbors, enter the **show ip bgp neighbors** command.

```
NetIron# show ip bgp neighbors
    Total number of BGP Neighbors: 6
1   IP Address: 50.50.50.10, AS: 20 (EBGP), RouterID: 10.10.10.20, VRF: default
    State: ESTABLISHED, Time: 0h0m18s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 34 seconds, HoldTimer Expire in 163 seconds
       Minimum Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
       GracefulRestartCapability: Received
           Restart Time 120 sec, Restart bit 0
           afi/safi 1/1, Forwarding bit 0
       GracefulRestartCapability: Sent
           Restart Time 120 sec, Restart bit 0
           afi/safi 1/1, Forwarding bit 1
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
```

The text in bold is the BGP4 restart information for the specified neighbor.

Displaying AS4 details

This section describes the use of the following **show** commands, which produce output that includes information about AS4s. Information that reflects AS4s appears in **bold**.

- **show ip bgp neighbor** shows whether the AS4 capability is enabled.
- **show ip bgp attribute-entries** shows AS4 path values and extended community values.
- **show ip bgp** shows the route entries with two and AS4 path information.
- **show ip extcommunity-list** shows the members of the extended community.
- **show route-map** shows the presence of any AS4 configuration data.
- **show ip as-path-access-lists** shows the presence of any AS4 configuration data.
- **show ip bgp config** shows the presence of any AS4 configuration data.

Route entries with four-byte path information

The **show ip bgp** command without of any optional parameters display AS4 path information, as indicated by the bold text in this example.

```
NetIron-mu2(config)#show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  47.1.1.0/24        192.168.1.5     1      100    0      90000 100 200 65535
65536 65537 65538 65539 75000
```

Syntax: **show ip bgp**

Current AS numbers

To display current AS numbers, use the **show ip bgp neighbors** command at any level of the CLI.

```
NetIron-mu2#show ip bgp neighbors
  neighbors              Details on TCP and BGP neighbor connections
    Total number of BGP Neighbors: 1
1   IP Address: 192.168.1.1, AS: 7701000 (IBGP), RouterID: 192.168.1.1, VRF:
default-vrf
    State: ESTABLISHED, Time: 0h3m33s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 49 seconds, HoldTimer Expire in 177 seconds
    Minimal Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       0       5         0            0
       Received: 1       1       5         0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: 0h3m33s    ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  unicast  capability
      Peer configured for IPV4 unicast  Routes
    Neighbor AS4 Capability Negotiation:
      Peer Negotiated AS4  capability
      Peer configured for AS4  capability
    As-path attribute count: 1
    Outbound Policy Group:
       ID: 1, Use Count: 1
    TCP Connection state: ESTABLISHED, flags:00000044 (0,0)
     Maximum segment size: 1460
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:   148, Received: 203
       Local host:  192.168.1.2, Local  Port: 179
       Remote host: 192.168.1.1, Remote Port: 8041
       ISentSeq:    1656867  SendNext:    1657016  TotUnAck:          0
       TotSent:         149  ReTrans:          19  UnAckSeq:    1657016
       IRcvSeq:     1984547  RcvNext:     1984751  SendWnd:       64981
       TotalRcv:        204  DupliRcv:        313  RcvWnd:        65000
       SendQue:           0  RcvQue:            0  CngstWnd:       5840
```

Syntax: show ip bgp neighbors

The information related to AS4s is highlighted in bold text.

Attribute entries

Use the **show ip bgp attribute-entries** command to see AS4 path values and extended community values, as the following example illustrates.The extended community values in bold reflect AS4s.

```
NetIron-mu2#show ip bgp attribute-entries
        Total number of BGP Attribute Entries: 18 (0)
1       Next Hop  :192.168.1.6        Metric   :1              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: SOO 300000:3
        AS Path   :90000 80000 (length 11)
        Address: 0x10e4e0c4  Hash:489 (0x03028536), PeerIdx 0
        Links: 0x00000000, 0x00000000, nlri: 0x10f4804a
        Reference Counts: 1:0:1, Magic: 51
2       Next Hop  :192.168.1.5        Metric   :1              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 200000:2
        AS Path   :90000 75000 (length 11)
        Address: 0x10e4e062  Hash:545 (0x0301e8f6), PeerIdx 0
        Links: 0x00000000, 0x00000000, nlri: 0x10f47ff0
        Reference Counts: 1:0:1, Magic: 49
```

Syntax: show ip bgp attribute-entries

Extended community

Support for AS4s is reflected in the values for extended community, as shown by the bold text in this output for the **show ip extcommunity-list** command.

```
NetIron-mu2#show ip extcommunity-list
ip extcommunity access list 1:
  permit RT 100000:123 SOO 150000:456
mu2#show route-map
route-map test permit 1
 match extcommunity 1
 set ip next-hop 192.168.1.15
route-map test permit 2
 set ip next-hop 192.168.1.101
```

Syntax: show ip extcommunity-list

AS-path prepend and extended community information

The AS-path prepend and extended community information is shown in this example of the **show route-map** command.

```
NetIron(config)# show route-map
route-map test permit 1
 match ip address  1
 set as-path prepend  75000
 set extcommunity RT 100000:123
 set extcommunity SOO 150000:456
route-map test permit 2
 match ip address  2
 set as-path prepend 80000
```

Syntax: show route-map [<*name*>]

The optional <*name*> parameter lets you name a specific route.

Running configuration

AS4s appear in the display of a running configuration, as shown.

```
NetIron(config-bgp)# show ip bgp config
Current BGP configuration:
router bgp
 local-as 7701000
 confederation identifier 120000
 confederation peers 80000
 neighbor 192.168.1.2 remote-as 80000
```

Access lists that contain AS4s

AS4s that exist in access lists are displayed by the command, as shown.

```
NetIron-mu2# show ip as-path-access-lists
ip as-path access list abc: 1 entries
     seq 10 permit _75000_
ip as-path access list def: 1 entries
     seq 5 permit _80000_
```

Formats of AS4s in show command output

To display the asdot and asdot+ notation for AS4s, enter the **as-format asdot** or **as-format asdot+** commands before you enter the **show ip bgp** command..

```
NetIron-mu2(config)# as-format asdot
NetIron-mu2(config)# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP,? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*> 47.1.1.0/24        192.168.1.5     1      100    0      1.24464 100 200 655
5 1.0 1.1 1.2 1.3 1.9464 ?
```

Syntax: as-format asdot

```
NetIron-mu2(config)# as-format asdot+
NetIron-mu2(config)# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  47.1.1.0/24        192.168.1.5     1      100    0      1.24464 0.100 0.200
0.65535 1.0 1.1 1.2 1.3 1.9464?
```

Syntax: as-format asdot

Displaying route-map continue clauses

This section contains examples of route-map continuation clauses. Both the route map and the routes to which it applies are described.

This example is a simple illustration of route-map continue clauses. If the match clause of either route map instance 5 or 10 matches, the route map traversal continues at instance 100.

```
route-map test permit 5
 match community my_community1
 set comm-list delete my_community1
 continue 100
route-map test permit 10
 match community my_community2
 set comm-list delete my_community2
 continue 100
route-map test permit 100
 match as-path my_aspath
 set community 1234:5678 additive
```

The following example shows the route map “test.” The **show ip bgp route** output shows the consequences of the action in instance 1 (set weight = 10); instance 2 (metric becomes 20); and instance 5 (prepend as_path 300).

```
NetIron(config-routemap test)# show route-map test
route-map test permit 1
 set weight 10
 continue 2
route-map test permit 2
 set metric 20
 continue 3
route-map test permit 3
 set community 10:20
 continue 4
route-map test permit 4
 set community 30:40
 continue 5
route-map test permit 5
 set as-path prepend 300
 continue 6

NetIron(config-routemap test)# show ip bgp route
Total number of BGP Routes: 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      8.8.8.0/24         8.8.8.3         20         100        10     BE
         AS_PATH: 300 200
```

Syntax: show route-map <*map-name*>

The <*map-name*> is the name of the route map.

Syntax: show ip bgp route

In the following example, the continue clause of instance 1 has been changed so that program flow jumps to instance 5. The resulting BGP4 route only has the weight updated and as-path prepended. These changes show route-map <*route name*>

Syntax: route-map

Syntax: [**no**] **continue** <*instance number*>

Syntax: show ip bgp route

In this example, a match clause has been added to instance 8. Because the match clause of instance 8 does not get fired, the search for the next instance continues to the end of the route-map. The set statements set the weight to 10, prepend 300, prepend 100 to the as-path, set the community to none, and set the local preference to 70. The results of this route-map traversal appear in the output of the **show ip bgp route** command.

```
NetIron(config-routemap test)# show route-map test
route-map test permit 1
 set weight 10
 continue 5
route-map test permit 2
 set metric 20
 continue 3
route-map test permit 3
 set community 10:20
 continue 4
route-map test permit 4
 set community 30:40
 continue 5
route-map test permit 5
 set as-path prepend 300
 continue 6
route-map test permit 6
 set as-path prepend 100
 continue 7
route-map test permit 7
 set community none
 set local-preference 70
 continue 8
route-map test deny 8
 match metric 60
 set metric 40
 continue 9
NetIron(config-routemap test)# show ip bgp route
Total number of BGP Routes: 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      8.8.8.0/24         8.8.8.3         0          70         10     BE
         AS_PATH: 100 300 200
```

Syntax: show route-map

Syntax: show ip bgp route

For this example, an existing route map is displayed by the **show route-map** command, then the addition of instance 8 adds a deny parameter but no match clause. As a result, no incoming routes are accepted (see last line of the show output).

```
NetIron(config-routemap test)#show route-map test
route-map test permit 1
 set weight 10
 continue 5
route-map test permit 2
 set metric 20
 continue 3
route-map test permit 3
 set community 10:20
 continue 4
route-map test permit 4
 set community 30:40
 continue 5
route-map test permit 5
 set as-path prepend 300
 continue 6
route-map test permit 6
 set as-path prepend 100
 continue 7
route-map test permit 7
 set community none
 set local-preference 70
 continue 8
NetIron(config-routemap test)#route-map test deny 8
NetIron(config-routemap test)#set metric 40
NetIron(config-routemap test)#continue 9
NetIron(config-routemap test)#show ip bgp route
```

BGP Routing Table is empty

Syntax: show route-map <*map-name*>

Configuring IP Multicast Protocols 27

Overview

The following displays the IP Multicast features supported by PowerConnect B-MLXe.

- IGMP (V1 and V2)
- IGMP (V1 and V2) Snooping
- IGMP v2 Fast-Leave
- IGMP v3
- IGMP v3 Snooping
- Multicast Routing PIM
- DVMRP
- PMRI
- PIM-SSM
- Multicast over Multi-VRF
- IP Multicast Boundaries
- PIM Dense
- PIM Sparse
- Modifying the TTL threshold
- Multicast Source Discovery Protocol (MSDP)
- MSDP Mesh Groups
- MSDP Anycast RP
- PIM Anycast RP
- Static Multicast Routes
- Optimization of Multicast Replication and Platform Independence
- Concurrent Support for Multicast Routing and Snooping
- Modifying the Prune Wait Timer
- Configuring PIM-SM (*,g) Forwarding
- IPv6 Support

This chapter describes how to configure devices for the following IP multicast protocol and versions:

- Internet Group Management Protocol (IGMP) V1 and V2
- Protocol Independent Multicast Dense mode (PIM DM) V1 (draft-ietf-pim-dm-05) and V2 (draft-ietf-pim-v2-dm-03)
- PIM Sparse mode (PIM SM) V2 (RFC 2362)
- Distance Vector Multicast Routing Protocol (DVMRP) V2 (RFC 1075)

NOTE
Each multicast protocol uses IGMP. IGMP is automatically enabled on an interface when you configure PIM or DVMRP, and is disabled on the interface if you disable PIM or DVMRP.

Overview of IP multicasting

Multicast protocols allow a group or channel to be accessed over different networks by multiple stations (clients) for the receipt and transmission of multicast data.

Distribution of stock quotes, video transmissions such as news services and remote classrooms, and video conferencing are all examples of applications that use multicast routing.

Dell devices support two multicast routing protocols—Distance Vector Multicast Routing Protocol (DVMRP) and Protocol-Independent Multicast (PIM) protocol, along with the Internet Group Membership Protocol (IGMP).

PIM and DVMRP are broadcast and pruning multicast protocols that deliver IP multicast datagrams. These protocols employ reverse path lookup check and pruning to allow source-specific multicast delivery trees to reach all group members. DVMRP and PIM build a different multicast tree for each source and destination host group.

Both DVMRP and PIM can concurrently operate on different ports of a device. The CAM can hold up to 1535 IPv4 multicast entries.

Multicast terms

The following terms are commonly used in discussing multicast-capable devices. These terms are used throughout this chapter:

Node: Refers to a device.

Root Node: The node that initiates the tree building process. It is also the device that sends the multicast packets down the multicast delivery tree.

Upstream: Represents the direction from which a device receives multicast data packets. An upstream device is a node that sends multicast packets.

Downstream: Represents the direction to which a device forwards multicast data packets. A **downstream** device is a node that receives multicast packets from upstream transmissions.

Group Presence: Means that a multicast group has been learned from one of the directly connected interfaces. Members of the multicast group are present on the device.

Intermediate nodes: Devices that are in the path between source devices and leaf devices.

Leaf nodes: Devices that do not have any downstream devices.

Multicast Tree: A unique tree is built for each source group (S,G) pair. A multicast tree is comprised of a root node and one or more nodes that are leaf or intermediate nodes.

Changing global IP multicast parameters

The following sections apply to PIM-DM, PIM-SM, IGMP, and DVMRP.

Concurrent support for multicast routing and snooping

Multicast routing and multicast snooping instances work concurrently on the same device. For example, you can configure PIM routing on certain VEs interfaces and snooping on other VEs or VLANs. The limitation is that either multicast snooping or routing can be enabled on a VE interface or VLAN, but not on both. This is because all of the multicast data and control packets (IGMP, PIM) received on the snooping VLAN are handled by multicast snooping and do not reach the multicast routing component. Similarly, any multicast data or control packets received on a VE interface enabled with PIM or DVMRP routing are handled by the PIM or DVMRP routing component and are not seen by the IGMP or PIM snooping component.

The following considerations apply when configuring concurrent operation of Multicast Routing and Snooping.

1. Either multicast snooping or routing can be enabled on a VE or VLAN but not both.
2. Snooping can be enabled globally (**ip multicast** <*active* | *passive*>) as can multicast routing (**ip multicast-routing**).
3. The global snooping configuration is inherited by all current VLANs that are not enabled for multicast routing.
4. The global snooping configuration is also inherited by all new VLANs. Enabling multicast routing on a newly created VLAN or VE automatically disables snooping on the VLAN or VE.
5. When a VLAN-level snooping is configured, it is displayed.

Defining the maximum number of DVMRP cache entries

You can use the following run-time command to define the maximum number of repeated DVMRP traffic being sent from the same source address and being received by the same destination address. The maximum number of multicast cache entries for DVMRP can be defined for the default VRF using the following command.

```
NetIron(config)# router dvmrp
NetIron(config-dvmrp-router)# max-mcache 500
```

Syntax: [no] max-mcache <*num*>

The <*num*> variable specifies the maximum number of multicast cache entries for DVMRP. If not defined by this command, the maximum value is determined by available system resources.

The maximum number of multicast cache entries for DVMRP for a specified Virtual Routing Instance (VRF) can be defined using the following command.

```
NetIron(config)# router dvmrp vrf vpn1
NetIron(config-dvmrp-router-vrf-vpn1)# max-mcache 500
```

Syntax: [no] router dvmrp [vrf <*vrf-name*>]

Syntax: [no] max-mcache <*num*>

The **vrf** parameter specified with the **router dvmrp** command allows you to configure the **max-mcache** command for a virtual routing instance (VRF) specified by the variable <*vrf-name*>.

The <*num*> variable specifies the maximum number of multicast cache entries for DVMRP in the specified VRF. If not defined by this command, the maximum value is determined by available system resources

Defining the maximum number of DVMRP routes

You can use the following run-time command to set the maximum number of DVMRP routes. The maximum number of DVMRP routes can be defined for the default VRF using the following command.

```
NetIron(config)# router dvmrp
NetIron(config-dvmrp-router)# max-route 500
```

The maximum number of DVMRP routes can be defined for a specified VRF using the following command.

```
NetIron(config)# router dvmrp vrf vpn1
NetIron(config-dvmrp-router-vrf-vpn1)# max-route 500
```

Syntax: [no] router dvmrp [vrf <*vrf-name*>]

Syntax: [no] max-route <*num*>

The **vrf** parameter specified with the **router dvmrp** command allows you to configure the **max-route** command for a virtual routing instance (VRF) specified by the variable <*vrf-name*>.

The <*num*> parameter specifies the maximum number of DVMRP routes. If not defined by this command, the maximum value is determined by available system resources.

Defining the maximum number of PIM cache entries

You can use the following run-time command to define the maximum number of repeated PIM traffic being sent from the same source address and being received by the same destination address. To define this maximum for the default VRF, enter the following commands.

```
NetIron(config)# router pim
NetIron(config-pim-router)# max-mcache 999
```

Syntax: [no] max-mcache <*num*>

The <*num*> variable specifies the maximum number of multicast cache entries for PIM in the default VRF. If not defined by this command, the maximum value is determined by available system resources.

To define the maximum number of PIM Cache entries for a specified VRF, use the following command.

```
NetIron(config)# router pim vrf vpn1
NetIron(config-pim-router-vrf-vpn1)# max-mcache 999
```

Syntax: [no] router pim [vrf <*vrf-name*>]

Syntax: [no] max-mcache <*num*>

The **vrf** parameter specified with the **router pim** command allows you to configure the **max-mcache** command for a virtual routing instance (VRF) specified by the variable <*vrf-name*>.

The <*num*> variable specifies the maximum number of multicast cache entries for PIM in the specified VRF. If not defined by this command, the maximum value is determined by available system resources.

Defining the maximum number of multicast VRF CAM entries

To use a run time command to set the maximum values for multicast VRF CAM entries for all VRFs or for a specified VRF.

Defining the maximum number of multicast VRF CAM entries for all VRFs

You can use the following run-time command to define the maximum number of multicast VRF CAM entries by entering a command such as the following.

```
NetIron(config)# ip multicast-max-all-vrf-cam 3072
```

Syntax: [no] ip multicast-max-all-vrf-cam <*num*>

The <*num*> variable specifies the maximum number of multicast VRF CAM entries for all VRFs. This setting does not effect the default VRF. The maximum possible value is 8000 and the default value is 2048.

Defining the maximum number of multicast VRF CAM entries for a specified VRF

You can use the following run-time command to define the maximum number of multicast VRF CAM entries for a specified VRF by entering commands such as the following.

```
NetIron(config)# ip vrf vpn1
NetIron(config-vrf-vpn1)# ip multicast-max-cam 3072
```

Syntax: [no] ip vrf <*vrf-name*>

Syntax: [no] ip multicast-max-cam <*num*>

The **ip vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

The <*num*> variable specifies the maximum number of multicast VRF CAM entries for the specified VRF. This setting can be any number up to the limit set using the **ip multicast-max-all-vrf-cam** command .

Defining the maximum number of IGMP group addresses

You can use the following run-time command to set the maximum number of IGMP addresses for the default VRF or for a specified VRF. To define this maximum for the default VRF, enter the following command.

```
NetIron(config)# ip igmp max-group-address 1000
```

Syntax: [no] ip igmp max-group-address <*num*>

The <*num*> variable specifies the maximum number of IGMP group addresses you want to make available for the default VRF. If not defined by this command, the maximum value is determined by available system resources.

To define this maximum for a specified VRF, enter the following commands.

```
NetIron(config)# ip vrf vpn1
NetIron(config-vrf-vpn1)# ip igmp max-group-address 1000
```

Syntax: [no] ip vrf <*vrf-name*>

Syntax: [no] ip igmp max-group-address <*num*>

The **ip vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

The <*num*> parameter specifies the number of IGMP group addresses that you want to make available for the specified VRF. If not defined by this command, the maximum value is determined by available system resources.

Changing IGMP V1 and V2 parameters

IGMP allows Dell devices to limit the multicast of IGMP packets to only those ports on the device that are identified as IP Multicast members.

The device actively sends out host queries to identify IP Multicast groups on the network, inserts the group information in an IGMP packet, and forwards the packet to IP Multicast neighbors.

The following IGMP V1 and V2 parameters apply to PIM and DVMRP:

- **IGMP query interval** – Specifies how often the PowerConnect queries an interface for group membership. Possible values are 2 – 3600. The default is 125.
- **IGMP group membership time** – Specifies how many seconds an IP Multicast group can remain on a PowerConnect interface in the absence of a group report. Possible values are 5 – 26000. The default is 260.
- **IGMP maximum response time** – Specifies how many seconds the device will wait for an IGMP response from an interface before concluding that the group member on that interface is down and removing the interface from the group. Possible values are 1 – 25. The default is 10.

To change these parameters, you must first enable IP multicast routing by entering the following CLI command at the global CLI level.

```
NetIron(config)# ip multicast-routing
```

Syntax: [no] ip multicast-routing

NOTE
You must enter the **ip multicast-routing** command before changing the global IP Multicast parameters. Otherwise, the changes do not take effect and the software uses the default values. Also, entering **no ip multicast-routing** will reset all parameters to their default values.

Modifying IGMP (V1 and V2) query interval period

The IGMP query interval period defines how often a device will query an interface for group membership. Possible values are 2 – 3600 seconds and the default value is 125 seconds.

To modify the default value for the IGMP (V1 and V2) query interval, enter the following.

```
NetIron(config)# ip igmp query-interval 120
```

Syntax: [no] ip igmp query-interval <*num*>

The <*num*> variable specifies the number of seconds and can be a value from 2 - 3600.

The default value is 125.

Modifying IGMP (V1 and V2) membership time

Group membership time defines how long a group will remain active on an interface in the absence of a group report. Possible values are from 5 – 26000 seconds and the default value is 260 seconds.

To define an IGMP (V1 and V2) membership time of 240 seconds, enter the following.

```
NetIron(config)# ip igmp group-membership-time 240
```

Syntax: [no] ip igmp group-membership-time <*num*>

The <*num*> variable specifies the number of seconds and can be a value from 5 - 26000.

The default value is 260.

Modifying IGMP (V1 and V2) maximum response time

Maximum response time defines how long the PowerConnect will wait for an IGMP (V1 and V2) response from an interface before concluding that the group member on that interface is down and removing the interface from the group. Possible values are 1 – 25. The default is 10.

To change the IGMP (V1 and V2) maximum response time, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# ip igmp max-response-time 8
```

Syntax: [no] ip igmp max-response-time <*num*>

The <*num*> variable specifies the number of seconds and can be a value from 1 – 25. The default is 10.

Security Enhancement for IGMP

A security enhancement has been made to IGMPv2 to adhere to the following recommendation of RFC 2236: “Ignore the Report if you cannot identify the source address of the packet as belonging to a subnet assigned to the interface on which the packet was received.”

NOTE
When used in applications such as IP-TV (or any multicast application in general), the administrator should ensure that the set-top box (or multicast client) is configured on the same subnet as the v.e. configured on the device. This is typically the case but is emphasized here to ensure correct operation. Without this configuration, IGMP messages received by the device are ignored which causes an interruption in any multicast traffic directed towards the set-top box (multicast client).

Support for Multicast Multi-VRF

Multicast Multi-VRF support for the PowerConnect includes the following:

- **Static Mroute** – As described in "Configuring a static multicast route within a VRF" on page 1220 you can configure static multicast route from within a specified VRF.
- **PIM (PIM-SM and PIM-DM)** – The procedure for configuring PIM within a VRF instance is described in "Enabling PIM for a specified VRF" on page 1157 and "Enabling PIM Sparse for a specified VRF" on page 1166.
- **DVMRP** – The procedure for configuring DVMRP within a VRF instance is described in "Enabling DVMRP for a specified VRF" on page 1208.

System max parameter changes

Several changes to the system max commands have been made in support of Multicast Multi-VRF. That includes retiring the following system max commands:

system-max multicast-route

system-max dvmrp-mcache

system-max dvmrp-route

system-max pim-mcache

system-max igmp-max-group-address

These commands which require a system reload to take effect have been replaced by the following runtime commands:

ip max-mroute – This command replaces the **system-max multicast-route** command.

max-mcache – This command described in "Defining the maximum number of DVMRP cache entries" on page 1141 replaces the **system-max dvmrp-mcache** command.

max-mroute – This command described in "Defining the maximum number of DVMRP routes" on page 1142 replaces the **system-max dvmrp-route** command.

max-mcache – This command described in "Defining the maximum number of PIM cache entries" on page 1142 replaces the **system-max pim-mcache** command.

ip igmp max-group-address – This command described in "Defining the maximum number of IGMP group addresses" on page 1143 replaces the **system-max igmp-max-group-address** command.

NOTE
If the retired system-max commands are used, the new runtime commands will be substituted in the running config.

Additionally, you can set a maximum value for multicast VRF CAM entries as described in "Defining the maximum number of multicast VRF CAM entries" on page 1143.

show and clear command support

The following show and clear commands have been introduced or enhanced to support Multicast Multi-VRF:

- "clear ip dvmrp [vrf <vrf-name>] cache"
- "show ip dvmrp [vrf <vrf-name>] interface"
- "show ip dvmrp [vrf <vrf-name>] mcache"
- "clear ip igmp [vrf <vrf-name>] cache"
- "clear ip igmp [vrf <vrf-name>] traffic"
- "show ip igmp [vrf <vrf-name>] group [<group-address> [detail] [tracking]]"
- "show ip igmp [vrf <vrf-name>] interface [ve <number> | ethernet <slot/port> | pos <slot/port> | tunnel <num>]"
- "show ip igmp [vrf <vrf-name>] settings"
- "show ip igmp [vrf <vrf-name>] traffic"

Adding an interface to a multicast group

You can manually add an interface to a multicast group. This is useful in the following cases:

- Hosts attached to the interface are unable to add themselves as members of the group using IGMP.
- There are no members for the group attached to the interface.

When you manually add an interface to a multicast group, the device forwards multicast packets for the group but does not itself accept packets for the group.

You can manually add a multicast group to individual ports only. If the port is a member of a virtual routing interface, you must add the ports to the group individually.

To manually add a port to a multicast group, enter a command such as the following at the configuration level for the port.

```
NetIron(config-if-e10000-1/1)# ip igmp static-group 224.2.2.2
```

This command adds port 1/1 to multicast group 224.2.2.2.

To add a port that is a member of a virtual routing interface to a multicast group, enter a command such as the following at the configuration level for the virtual routing interface.

```
NetIron(config-vif-1)# ip igmp static-group 224.2.2.2 ethernet 5/2
```

This command adds port 5/2 in virtual routing interface 1 to multicast group 224.2.2.2.

Syntax: [no] ip igmp static-group <*ip-addr*> [ethernet <*slot*>/<*portnum*>]

The <*ip-addr*> parameter specifies the group number.

The **ethernet** <*slot*>/<*portnum*> parameter specifies the port number. Use this parameter if the port is a member of a virtual routing interface, and you are entering this command at the configuration level for the virtual routing interface.

Manually added groups are included in the group information displayed by the following commands:

- show ip igmp group
- show ip pim group

Multicast non-stop routing

Multicast non-stop routing (NSR) provides hitless upgrade and switchover support for all IPv4 multicast, including default and non-default VRFs for IPv4 PIM-DM, PIM-SM, and PIM-SSM. Multicast NSR is not supported for IPv6 multicast and DVMRP. The software multicast state is kept in sync between the active and standby MPs. As the PowerConnect system enters a hitless upgrade or switchover state, the standby MP will take over as the new active MP. The new active MP will carry a pre-installed multicast state that was originally supported by the previous MP. The new active MP will revalidate the pre-installed multicast state, and pick up any new changes as needed before marking the multicast state as operational. When the LP is ready to complete the hitless upgrade or switchover process, the operational multicast state will be downloaded to the LP CPU. When the LP resets, and the outage of the LP CPU occurs, pre-existing hardware forwarding multicast traffic will continue to flow without disruption, and the hardware multicast forwarding state is retained in the LP hardware.

Multicast NSR is globally enabled across all VRFs by configuring the **ip multicast-nonstop-routing** command. For more information on configuring the **ip multicast-nonstop-routing** command, refer to "Configuring multicast non-stop routing" on page 1148.

NOTE
During hitless reload, if any changes occur to the existing multicast forwarding records, then multicast receivers of the same forwarding records may see traffic loss.

Configuration considerations

- Multicast NSR is not supported for IPv6 multicast, DVMRP, and layer 2 multicast.
- When multicast NSR is turned on, unicast routing must be protected by NSR or graceful restart on all multicast VRFs.
- Any multicast flow that does not have a hardware CAM entry programmed prior to hitless upgrade or switchover will not be protected under multicast NSR. The multicast entry for such a flow shall be recreated upon the completion of the NSR process.

Configuring multicast non-stop routing

To globally enable multicast non-stop routing for all VRFs, enter the **ip multicast-nonstop-routing** command on the CLI as shown in the example below.

```
NetIron(config)#ip multicast-nonstop-routing
```

Syntax: ip multicast-nonstop-routing

During a hitless upgrade and switchover on the MP, the following syslog message is generated on the CLI.

```
Feb  3 14:09:58 Mcastv4 detected MP switchover, set switchover in progress to TRUE
Feb  3 14:10:07 Mcastv4 confirms unicast RTM is ready
Feb  3 14:10:07 Mcastv4 switchover done, set switchover in progress mode to FALSE
```

The syslog message displayed above shows the state transition of multicast NSR as the standby MP takes over as the active MP. The multicast data traffic will continue to flow during state transition.

Displaying the multicast NSR status

To display the multicast NSR status, enter the following command.

```
NetIron#show ip pim nsr
Global Mcast NSR Status
  NSR: ON
  Switchover In Progress Mode: FALSE
  Dy-Sync Postpone Flag: FALSE
```

The following table displays the output from the **show ip pim nsr** command.

TABLE 179 Output from the show ip pim nsr

This field...	Displays...
NSR	The NSR field indicates if the **ip multicast-nonstop-routing** command is enabled (ON) or disabled (OFF).
Switchover in Progress Mode	The Switchover in Progress Mode field indicates if the multicast traffic is in the middle of a switchover (displaying a TRUE status), or not (displaying a FALSE status).
Dy-Sync Postpone Flag	After the current switchover or hitless upgrade is complete, an update to the batched dy-sync may or may not need posting.

Displaying counter and statistic information for multicast NSR

To display multicast NSR counter and statistic information from the MP, enter the following command.

```
NetIron#show ip pim counter nsr
Mcache sync (entity id: 203)
  pack: 0
  unpack: 0
  ack: 0
RPset sync (entity id: 201)
  pack: 0
  unpack: 0
  ack: 0
BSR status (entity id: 202)
  pack: 1
  unpack: 0
  ack: 1
```

Syntax: show ip pim [vrf<*vrf_name*>] counter nsr

The **vrf** parameter allows you to display IP PIM counters for the VRF instance specified by the <*vrf-name*> variable.

The following table displays the output from the **show ip pim counter nsr** command.

TABLE 180 Output from the **show ip pim counter nsr** command

This field...	Displays...
Mcache sync	The mcache NSR sync queue that carries the NSR sync message for mcache updates.
pack	The number of NSR sync messages that are packed from current MP to the other MP.
unpack	The number of NSR sync messages that are received and unpacked by the current MP.
ack	The number of NSR sync acknowledgement the current MP received.
RPset sync	The RPset sync queue that carries the NSR sync message for RPset update.
BSR status	The BSR status sync queue that carries the NSR sync message for BSR information update.

Passive Multicast Route Insertion (PMRI)

To prevent unwanted multicast traffic from being sent to the CPU, PIM Routing and Passive Multicast Route Insertion (PMRI) can be used together to ensure that multicast streams are only forwarded out ports with interested receivers and unwanted traffic is dropped in hardware on Layer 3 Switches. This feature does not apply to DVMRP traffic.

PMRI enables a Layer 3 switch running PIM Sparse to create an entry for a multicast route (e.g., (S,G)), with no directly attached clients or when connected to another PIM device (transit network).

When a multicast stream has no output interfaces, the Layer 3 Switch can drop packets in hardware if the multicast traffic meets either of the following conditions:

In PIM-SM:

- The route has no OIF *and*
- If directly connected source passed source RPF check *and* completed data registration with RP *or*
- If non directly connected source passed source RPF check.

In PIM-DM:

- The route has no OIF *and*
- passed source RPF check *and*
- Device has no downstream PIM neighbor.

If the OIF is inserted after the hardware-drop entries are installed, the hardware entries will be updated to include the OIFs.

NOTE
Disabling hardware-drop does not immediately take away existing hardware-drop entries, they will go through the normal route aging processing when the traffic stops.

Configuring PMRI

PMRI is enabled by default. To disable PMRI, enter commands such as the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# hardware-drop-disable
```

Syntax: [no] hardware-drop-disable

Displaying hardware-drop

Use the **show ip pim sparse** command to display if the hardware-drop feature has been enabled or disabled.

```
NetIron(config)#show ip pim sparse
Global PIM Sparse Mode Settings
  Hello interval        : 30          Neighbor timeout                    : 105
  Bootstrap Msg interval: 60          Candidate-RP Advertisement interval: 60
  Join/Prune interval   : 60          SPT Threshold                       : 1
  Inactivity interval   : 180         SSM Enabled                         : No
  Hardware Drop Enabled : Yes
  show ip pim sparse
```

IP multicast boundaries

The Multicast Boundary feature is designed to selectively allow or disallow multicast flows to configured interfaces.

The **ip multicast-boundary** command allows you to configure a boundary on PIM enabled interface by defining which multicast groups may not forward packets over a specified interface. This includes incoming and outgoing packets. By default, all interfaces that are enabled for multicast are eligible to participate in a multicast flow provided they meet the multicast routing protocol's criteria for participating in a flow.

Configuration considerations

The configuration considerations are as follows:

- Only one ACL can be bound to any interface.
- Normal ACL restrictions apply as to how many software ACLs can be created, but there is no hardware restrictions on ACLs with this feature.
- Creation of a static IGMP client is allowed for a group on a port that may be prevented from participation in the group on account of an ACL bound to the port's interface. In such a situation, the ACL would prevail and the port will not be added to the relevant entries.
- Either standard or extended ACLs can be used with the multicast boundary feature. When a standard ACL is used, the address specified is treated as a group address and NOT a source address.
- When a boundary is applied to an ingress interface, all packets destined to a multicast group that is filtered out will be dropped by software. Currently, there is no support to drop such packets in hardware.

- The **ip multicast-boundary** command may not stop clients from receiving multicast traffic if the filter is applied on the egress interface up-stream from RP.

Configuring multicast boundaries

Multicast boundaries can be configured for IPv4 or IPv6.

To define boundaries for PIM enabled interfaces, enter a commands such as the following.

```
NetIron(config)# interface ve 40
NetIron(config-vif-40)#ip multicast-boundary DellAccessList
```

Multicast boundaries can be configured for IPv6 as shown in the following.

```
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e1000-1/2)#ipv6 multicast-boundary DellAccessList
```

Syntax: [no] ip multicast-boundary <*acl-spec*>

Syntax: [no] ipv6 multicast-boundary <*acl-spec*>

Use the **acl-spec** parameter to define the number or name identifying an access list that controls the range of group addresses affected by the boundary.

Use the **no ip multicast boundary** command to remove the boundary on a PIM enabled interface.

The ACL, DellAccessList can be configured using standard ACL syntax. ACLs are described in Chapter 20, “Layer 2 Access Control Lists,” however, some examples of how ACLs can be used to filter multicast traffic are provided below:

Standard ACL to permit multicast traffic

To permit multicast traffic for group 225.1.0.2 and deny all other traffic, enter the following command.

```
NetIron(config)# access-list 10 permit host 225.1.0.2
NetIron(config)# access-list 10 deny any
```

Extended ACL to deny multicast traffic

To deny multicast data traffic from group 225.1.0.1 and permit all other traffic,

```
NetIron(config)# access-list 101 deny ip any host 225.1.0.1
NetIron(config)# access-list 101 permit ip any any
```

Extended ACL to permit multicast traffic

To permit multicast data traffic from source 97.1.1.50 for group 225.1.0.1 and deny all other traffic,

```
NetIron(config)# access-list 102 permit ip host 97.1.1.50 host 225.1.0.1
NetIron(config)# access-list 102 deny ip any any
```

Displaying multicast boundaries

To display multicast boundary information, use the **show ip pim interface** command.

```
NetIron# show ip pim interface
---------+------------+----+---+-----------------+---+---------+-------+-------
Interface|Local       |Mode|Ver|Designated Router |TTL|Multicast| VRF   | DR
         |Address     |    |   |Address       Port|Thr|Boundary |       | Prio
---------+------------+----+---+-----------------+---+---------+-------+-------
     v10   10.1.2.1       SM  V2  Itself             1  None     default  30
     v30  123.1.1.2       SM  V2  Itself             1  None
     v40  124.1.1.2       SM  V2  Itself             1  101
```

Syntax: show ip pim [vrf <*vrf-name*>] interface [ethernet <*slot*>/<*portnum*> | ve <*num*> | tunnel <*num*>]

The **vrf** option allows you to display multicast boundary information for the VRF instance identified by the <*vrf-name*> variable.

The **ethernet** <*port-number*> parameter specifies the physical port.

The **ve** <*num*> parameter specifies a virtual interface.

The **tunnel** <*num*> parameter specifies a GRE tunnel interface that is being configured. The GRE tunnel interface is enabled under the device PIM configuration.

PIM Dense

NOTE
This section describes the "dense" mode of PIM, described in RFC 3973. Refer to "PIM Sparse" on page 1163 for information about PIM Sparse.

PIM was introduced to simplify some of the complexity of the routing protocol at the cost of additional overhead tied with a greater replication of forwarded multicast packets. PIM is similar to DVMRP in that PIM builds source-routed multicast delivery trees and employs reverse path check when forwarding multicast packets.

There are two modes in which PIM operates: Dense and Sparse. The Dense Mode is suitable for densely populated multicast groups, primarily in the LAN environment. The Sparse Mode is suitable for sparsely populated multicast groups with the focus on WAN.

PIM primarily differs from DVMRP by using the IP routing table instead of maintaining its own, thereby being routing protocol independent.

Initiating PIM multicasts on a network

Once PIM is enabled on each device, a network user can begin a video conference multicast from the server on R1 as shown in Figure 154. When a multicast packet is received on a PIM-capable device interface, the interface checks its IP routing table to determine whether the interface that received the message provides the shortest path back to the source. If the interface does provide the shortest path back to the source, the multicast packet is then forwarded to all neighboring PIM devices. Otherwise, the multicast packet is discarded and a prune message is sent back upstream.

In Figure 154, the root node (R1) is forwarding multicast packets for group 229.225.0.1, which it receives from the server, to its downstream nodes, R2, R3, and R4. Device R4 is an intermediate device with R5 and R6 as its downstream devices. Because R5 and R6 have no downstream interfaces, they are leaf nodes. The receivers in this example are those workstations that are resident on devices R2, R3, and R6.

Pruning a multicast tree

As multicast packets reach these leaf devices, the devices check their IGMP databases for the group. If the group is not in the IGMP database of the device, the device discards the packet and sends a prune message to the upstream device. The device that discarded the packet also maintains the prune state for the source, group (S,G) pair. The branch is then pruned (removed) from the multicast tree. No further multicast packets for that specific (S,G) pair will be received from that upstream device until the prune state expires. You can configure the PIM Prune Timer (the length of time that a prune state is considered valid).

For example, in Figure 154 the sender with address 207.95.5.1 is sending multicast packets to the group 229.225.0.1. If a PIM device receives any groups other than that group, the device discards the group and sends a prune message to the upstream PIM device.

In Figure 155, device R5 is a leaf node with no group members in its IGMP database. Therefore, the device must be pruned from the multicast tree. R5 sends a prune message upstream to its neighbor device R4 to remove itself from the multicast delivery tree and install a prune state, as seen in Figure 155. Device 5 will not receive any further multicast traffic until the prune age interval expires.

When a node on the multicast delivery tree has all of its downstream branches (downstream interfaces) in the prune state, a prune message is sent upstream. In the case of R4, if both R5 and R6 are in a prune state at the same time, R4 becomes a leaf node with no downstream interfaces and sends a prune message to R1. With R4 in a prune state, the resulting multicast delivery tree would consist only of leaf nodes R2 and R3.

FIGURE 154 Transmission of multicast packets from the source to host group members

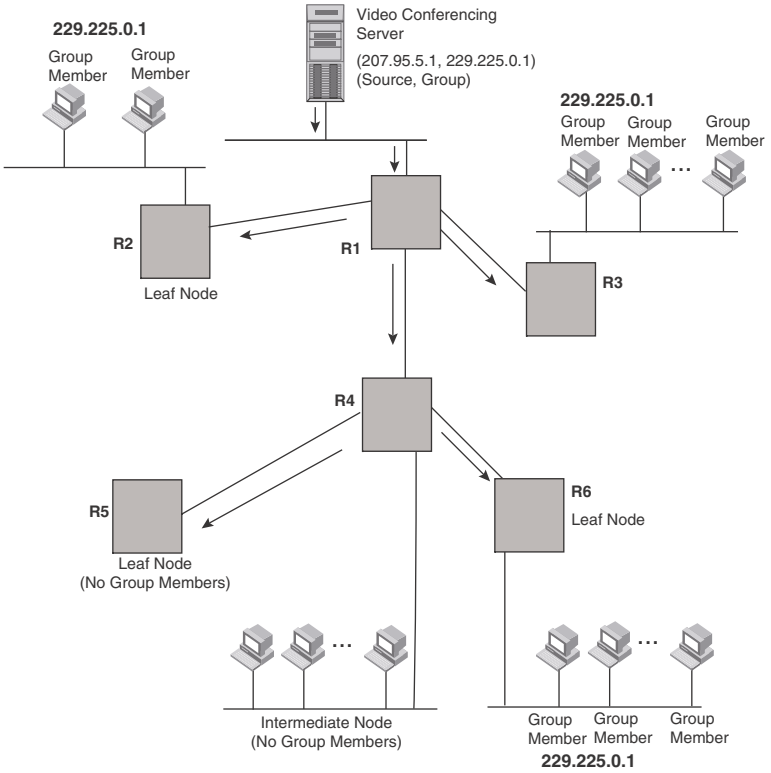
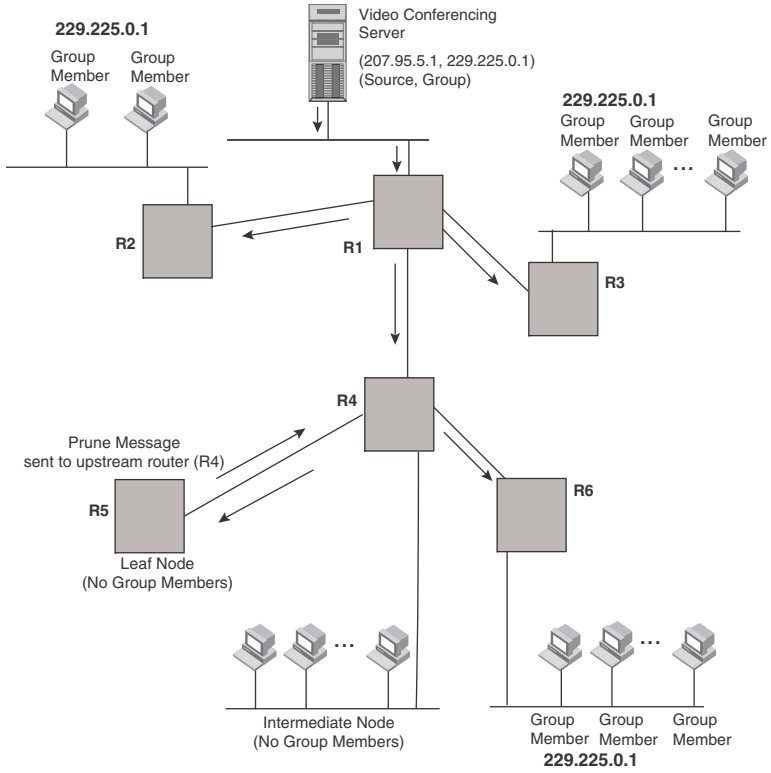


FIGURE 155 Pruning leaf nodes from a multicast tree



Grafts to a multicast tree

A PIM device restores pruned branches to a multicast tree by sending graft messages towards the upstream device. Graft messages start at the leaf node and travel up the tree, first sending the message to its neighbor upstream device.

In the example above, if a new 229.255.0.1 group member joins on device R6, which was previously pruned, a graft is sent upstream to R4. Since the forwarding state for this entry is in a prune state, R4 sends a graft to R1. Once R4 has joined the tree, R4 along with R6 once again receive multicast packets.

Prune and graft messages are continuously used to maintain the multicast delivery tree. No configuration is required on your part.

PIM DM versions

The device supports PIM DM V1 and V2. The default is V2. You can specify the version on an individual interface basis.

The primary difference between PIM DM V1 and V2 is the methods the protocols use for messaging:

- PIM DM V1 – uses the IGMP to send messages.
- PIM DM V2 – sends messages to the multicast address 224.0.0.13 (ALL-PIM-ROUTERS) with protocol number 103.

The CLI commands for configuring and managing PIM DM are the same for V1 and V2. The only difference is the command you use to enable the protocol on an interface.

NOTE
If you want to continue to use PIM DM V1 on an interface, you must change the version, then save the configuration.

NOTE
The note above does not mean you can run different PIM versions on devices that are connected to each other. The devices must run the same version of PIM. If you want to connect a device running PIM to a device that is running PIM V1, you must change the PIM version on the device to V1 (or change the version on the device to V2, if supported).

Configuring PIM DM

NOTE
This section describes how to configure the “dense” mode of PIM, described in RFC 1075. Refer to “Configuring PIM Sparse” on page 1165 for information about configuring PIM Sparse.

Enabling PIM on the device and an interface

By default, PIM is disabled. To enable PIM:

- Enable the feature globally.
- Configure the IP interfaces that will use PIM.

- Enable PIM locally on the ports that have the IP interfaces you configured for PIM.

Suppose you want to initiate the use of desktop video for fellow users on a sprawling campus network. All destination workstations have the appropriate hardware and software but the devices that connect the various buildings need to be configured to support PIM multicasts from the designated video conference server as shown in Figure 154 on page 1155.

PIM is enabled on each of the devices shown in Figure 154, on which multicasts are expected. You can enable PIM on each device independently or remotely from one of the devices with a Telnet connection. Follow the same steps for each device. All changes are dynamic.

Globally enabling and disabling PIM

To globally enable PIM, enter the following command.

```
NetIron(config)# device pim
```

Syntax: [no] router pim

NOTE
When PIM routing is enabled, the line rate for receive traffic is reduced by about 5%. The reduction occurs due to overhead from the VLAN multicasting feature, which PIM routing uses. This behavior is normal and does not indicate a problem with the device.

The [no] router pim command behaves in the following manner:

- Entering router pim command to enable PIM does not require a software reload.
- Entering a no router pim command removes all configuration for PIM multicast on a device (router pim level) only.

Enabling PIM for a specified VRF

To enable PIM for the VRF named “blue”, use the following commands.

```
NetIron(config)# router pim vrf blue
```

Syntax: [no] router pim [vrf <*vrf-name*>]

The vrf parameter allows you to configure PIM (PIM-DM and PIM-SM) on the virtual routing instance (VRF) specified by the <*vrf-name*> variable. All PIM parameters available for the default device instance are configurable for a VRF-based PIM instance.

The [no] router pim vrf command behaves in the following manner:

- Entering the router pim vrf command to enable PIM does not require a software reload.
- Entering a no router pim vrf command removes all configuration for PIM multicast on the specified VRF.

Enabling a PIM version

To enable PIM on an interface, globally enable PIM, then enable PIM on interface 3, enter the following commands.

```
NetIron(config)# router pim
NetIron(config)# int e 1/3
NetIron(config-if-e10000-1/3)# ip address 207.95.5.1/24
NetIron(config-if-e10000-1/3)# ip pim
NetIron(config-if-e10000-1/3)# write memory
NetIron(config-if-e10000-1/3)# end
```

Syntax: [**no**] **ip pim** [**version 1** | **2**]

The **version 1 | 2** parameter specifies the PIM DM version. The default version is 2.

If you have enabled PIM version 1 but need to enable version 2 instead, enter either of the following commands at the configuration level for the interface.

```
NetIron(config-if-e10000-1/1)# ip pim version 2
NetIron(config-if-e10000-1/1)# no ip pim version 1
```

To disable PIM DM on the interface, enter the following command.

```
NetIron(config-if-e10000-1/1)# no ip pim
```

Modifying PIM global parameters

PIM global parameters come with preset values. The defaults work well in most networks, but you can modify the following parameters if necessary:

- Neighbor timeout
- Hello timer
- Prune timer
- Prune wait timer
- Graft retransmit timer
- Inactivity timer

Modifying neighbor timeout

Neighbor timeout is the interval after which a PIM device will consider a neighbor to be absent. Absence of PIM hello messages from a neighboring device indicates that a neighbor is not present.

The default value is 105 seconds.

To apply a PIM neighbor timeout value of 360 seconds to all ports on the device operating with PIM, enter the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# nbr-timeout 360
```

Syntax: [**no**] **nbr-timeout** <*seconds*>

The default is 105 seconds. The range is 60-8000 seconds.

Modifying hello timer

This parameter defines the interval at which periodic hellos are sent out PIM interfaces. Devices use hello messages to inform neighboring devices of their presence. The default rate is 30 seconds.

To apply a PIM hello timer of 120 seconds to all ports on the device operating with PIM, enter the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# hello-timer 120
```

Syntax: [no] hello-timer <*10-3600*>

The default is 30 seconds.

Modifying prune timer

This parameter defines how long a PIM device will maintain a prune state for a forwarding entry.

The first received multicast interface is forwarded to all other PIM interfaces on the device. If there is no presence of groups on that interface, the leaf node sends a prune message upstream and stores a prune state. This prune state travels up the tree and installs a prune state.

A prune state is maintained until the prune timer expires or a graft message is received for the forwarding entry. The default value is 180 seconds.

To set the PIM prune timer to 90, enter the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# prune-timer 90
```

Syntax: [**no**] **prune-timer** <*seconds*>

The default is 180 seconds. The range is 60-3600 seconds.

Modifying the prune wait timer

The **prune-wait** command allows you to configure the amount of time a PIM device will wait before stopping traffic to neighbor devices that do not want the traffic. The value can be from zero to fifteen seconds. The default is three seconds. A smaller prune wait value reduces flooding of unwanted traffic.

A prune wait value of zero causes the PIM device to stop traffic immediately upon receiving a prune message. If there are two or more neighbors on the physical port, then the prune-wait command should not be used because one neighbor may send a prune message while the other sends a join message at the same time, or within less than three seconds.

To set the prune wait time to zero, enter the following commands.

```
NetIron(config)#router pim
NetIron(config-pim-router)#prune-wait 0
```

Syntax: [**no**] **prune-wait** <seconds>

The <*seconds*> can be 0 - 15. A value of 0 causes the PIM device to stop traffic immediately upon receiving a prune message. The default is 3 seconds.

To view the currently configured prune wait time, enter the **show ip pim dense** command as described in "Displaying basic PIM Dense configuration information" on page 1161.

Modifying graft retransmit timer

The graft retransmit timer defines the interval between the transmission of graft messages.

A graft message is sent by a device to cancel a prune state. When a device receives a graft message, the device responds with a Graft Ack (acknowledge) message. If this Graft Ack message is lost, the device that sent the graft message will resend it.

To change the graft retransmit timer from the default of 180 to 90 seconds, enter the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# graft-retransmit-timer 90
```

Syntax: [**no**] **graft-retransmit-timer** <*seconds*>

The default is 180 seconds. The range is from 60-3600 seconds.

Modifying inactivity timer

The device deletes a forwarding entry if the entry is not used to send multicast packets. The PIM inactivity timer defines how long a forwarding entry can remain unused before the device deletes it.

To apply a PIM inactivity timer of 90 seconds to all PIM interfaces, enter the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# inactivity-timer 90
```

Syntax: [**no**] **inactivity-timer** <*seconds*>

The default is 180 seconds. The range is from 60-3600 seconds.

Selection of shortest path back to source

By default, when a multicast packet is received on a PIM-capable interface in a multi-path topology, the interface checks its IP routing table to determine the shortest path back to the source. If the alternate paths have the same cost, the first alternate path in the table is picked as the path back to the source. For example, in the following example, the first four routes have the same cost back to the source. However, 137.80.127.3 is chosen as the path to the source since it is the first one on the list. The device rejects traffic from any port other than Port V11 on which 137.80.127.3 resides

```
Total number of IP routes: 19
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF Cost - Dist/Metric
     Destination     Gateway         Port       Cost      Type
       ..
     172.17.41.4     137.80.127.3    v11        2          O
     172.17.41.4     137.80.126.3    v10        2          O
     172.17.41.4     137.80.129.1    v13        2          O
     172.17.41.4     137.80.128.3    v12        2          O
     172.17.41.8     0.0.0.0         1/2        1          D
```

Failover time in a multi-path topology

When a port in a multi-path topology fails, multicast devices, depending on the routing protocol being used, take a few seconds to establish a new path, if the failed port is the input port of the downstream device.

Modifying the TTL threshold

The TTL threshold defines the minimum value required in a packet for it to be forwarded OUT of the interface AFTER the TTL has been decremented.

For example, if the TTL for an interface is set at 10, only those packets that enter with a TTL value of 11 or more are forwarded through the TTL-10 interface. With a default TTL threshold of 1, only packets ingressing with a TTL of 2 or greater are forwarded. The TTL threshold only applies to routed interfaces and is ignored by switched interfaces. Possible TTL values are 1 to 64. The default TTL value is 1.

To configure a TTL of 45, enter a command such as the following.

```
NetIron(config-if-e10000-3/24)# ip pim ttl-threshold 45
```

Syntax: [**no**] **ip pim ttl-threshold** <*1-64*>

Configuring a DR priority

The DR priority option lets you give preference to a particular device in the DR election process by assigning it a numerically higher DR priority. This value can be set for IPv4 and IPv6 interfaces. To set a DR priority higher than the default value of 1, use the **ip pim dr-priority** command as shown:

For IPv4.

```
NetIron(config-if-e10000-3/24)# ip pim dr-priority 50
```

For IPv6.

```
NetIron(config-if-e10000-3/24)# ipv6 pim dr-priority 50
```

Syntax: [**no**] **ip pim dr-priority** <*priority-value*>

Syntax: [**no**] *ipv6* **pim dr-priority** <*priority-value*>

The <*priority-value*> variable is the value that you want to set for the DR priority. Optional values are: 0 - 65535. The default value is 1.

The **no** option removes the command and sets the DR priority back to the default value of 1.

The following information may be useful for troubleshooting.

1. If more than one device has the same DR priority on a subnet (as in the case of default DR priority on all), the device with the numerically highest IP address on that subnet is elected as the DR.
2. The DR priority information is used in the DR election ONLY IF ALL the PIM devices connected to the subnet support the DR priority option. If there is at least one PIM device on the subnet that does not support this option, then the DR election falls back to the backwards compatibility mode in which the device with the numerically highest IP address on the subnet is declared the DR regardless of the DR priority values.

Displaying basic PIM Dense configuration information

To display PIM Dense configuration information, enter the following command at any CLI level.

```
NetIron(config)# show ip pim dense
Global PIM Dense Mode Settings
  Maximum Mcache            : 0        Current Count                    : 0
  Hello interval            : 30       Neighbor timeout                 : 105
  Join/Prune interval       : 60       Inactivity interval              : 180
  Graft Retransmit interval : 180      Prune Age                        : 180
  Hardware Drop Enabled     : Yes      Prune Wait Interval              : 3
  Route Precedence          : mc-non-default mc-default uc-non-default uc-default
---------+--------------+----+---+--------------------+---+---------+-------+-
Interface|Local        |Mode|Ver|  Designated Router  |TTL|Multicast|  VRF  | DR
         |Address      |    |   |Address         Port|Thr|Boundary |       | Prio
---------+--------------+----+---+--------------------+---+---------+-------+-
     e1/2 65.19.11.1       DM  V2  Itself                1 None      default 1
     e1/4 97.47.11.1       DM  V2  Itself                1 None      default 1
    e2/13 65.29.11.1       DM  V2  Itself               1 None      default 1
    e2/18 97.49.11.1       DM  V2  Itself               1 None      default 1
     p3/1 65.30.12.1       DM  V2  Itself               1 None      default 1
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **dense**

The **vrf** option allows you to display PIM dense configuration information for the VRF instance identified by the <*vrf-name*> variable.

This display shows the following information.

Table 0.1:

This field...	Displays...
Maximum Mcache	The maximum number multicast cache entries allowed on the device.
Current Count	The number of multicast cache entries currently used.
Hello interval	How frequently the device sends hello messages out the PIM dense interfaces.
Neighbor timeout	The interval after which a PIM device will consider a neighbor to be absent.
Graft or Retransmit interval	How interval between the transmission of graft messages.
Inactivity interval	How long a forwarding entry can remain unused before the device deletes it.
Join or Prune interval	How long a PIM device will maintain a prune state for a forwarding entry..
Prune Age	The number of packets the device sends using the path through the RP before switching to using the SPT path.
Hardware Drop Enabled	Displays Yes if the Passive Multicast Route Insertion feature is enable and No if it is not.
Prune Wait Interval	The amount of time a PIM device waits before stopping traffic to neighbor devices that do not want the traffic. The value can be from zero to three seconds. The default is three seconds.
Route Precedence	The route precedence configured to control the selection of routes based on the route types. There are four different types of routes: • Non-default route from the mRTM • Default route from the mRTM • Non-default route from the uRTM • Default route from the uRTM
Interface	The type of interface and the interface number.
Local Address	Indicates the IP address configured on the port or virtual interface.
Mode	Either DM for Dense Mode or SM for Sparse Mode.
Ver	The version of PIM Dense mode.
Designated Router	The IP address and port for the PIM designated device.
TTL Threshold	The minimum value required in a packet for it to be forwarded out of the interface.
Multicast Boundary	The multicast boundary enabled (if any) for PIM interfaces .
VRF	If the PIM Dense instance is configured within a VRF, this field will contain the name.
DR Prio	The priority of the designated device.

Displaying all multicast cache entries in a pruned state

You can use the following command to display all multicast cache entries that are currently in a pruned state and have not yet aged out.

```
NetIron(config-if-e10000-1/1)#show ip pim vrf med prune
Index Port    PhyPort SourceNet       Group           Nbr             Age
                                                                      sec
    1 v12     1/1     130.47.2.10     228.172.0.77    0.0.0.0         40
    2 v12     1/1     130.47.2.10     228.172.0.73    0.0.0.0         40
    3 v12     1/1     130.47.2.10     228.172.0.69    0.0.0.0         40
    4 v12     1/1     130.47.2.10     228.172.0.65    0.0.0.0         40
    5 v12     1/1     130.47.2.10     228.172.0.61    0.0.0.0         40
    6 v12     1/1     130.47.2.10     228.172.0.57    0.0.0.0         40
    7 v12     1/1     130.47.2.10     228.172.0.53    0.0.0.0         40
    8 v12     1/1     130.47.2.10     228.172.0.49    0.0.0.0         40
    9 v12     1/1     130.47.2.10     228.172.0.45    0.0.0.0         40
   ---------------------------------------
Total Prune entries: 9
```

Syntax: show ip pim [vrf <*vrf-name*>] prune

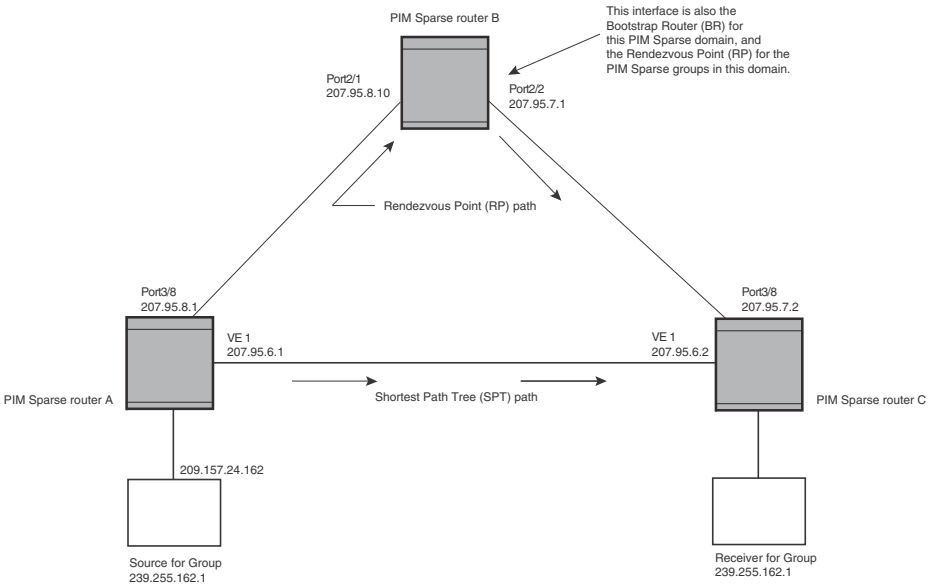
PIM Sparse

Dell devices support Protocol Independent Multicast (PIM) Sparse version 2. PIM Sparse provides multicasting that is especially suitable for widely distributed multicast environments. The Dell implementation is based on RFC 2362.

In a PIM Sparse network, a PIM Sparse device that is connected to a host that wants to receive information for a multicast group must explicitly send a join request on behalf of the receiver (host).

PIM Sparse devices are organized into domains. A PIM Sparse domain is a contiguous set of devices that all implement PIM and are configured to operate within a common boundary. Figure 156 shows a simple example of a PIM Sparse domain. This example shows three devices configured as PIM Sparse devices. The configuration is described in detail following the figure.

FIGURE 156 Example PIM Sparse domain



PIM Sparse device types

Devices that are configured with PIM Sparse interfaces also can be configured to fill one or more of the following roles:

- **PMBR** – A PIM device that has some interfaces within the PIM domain and other interface outside the PIM domain. PBMRs connect the PIM domain to the Internet.
- **BSR** – The Bootstrap Router (BSR) distributes RP information to the other PIM Sparse devices within the domain. Each PIM Sparse domain has one active BSR. For redundancy, you can configure ports on multiple devices as candidate BSRs. The PIM Sparse protocol uses an election process to select one of the candidate BSRs as the BSR for the domain. The BSR with the highest BSR priority (a user-configurable parameter) is elected. If the priorities result in a tie, then the candidate BSR interface with the highest IP address is elected. In the example in Figure 156, PIM Sparse device B is the BSR. Port 2/2 is configured as a candidate BSR.
- **RP** – The RP is the meeting point for PIM Sparse sources and receivers. A PIM Sparse domain can have multiple RPs, but each PIM Sparse multicast group address can have only one active RP. PIM Sparse devices learn the addresses of RPs and the groups for which they are responsible from messages that the BSR sends to each of the PIM Sparse devices. In the example in Figure 156, PIM Sparse device B is the RP. Port 2/2 is configured as a candidate Rendezvous Point (RP).

  To enhance overall network performance, the device uses the RP to forward only the first packet from a group source to the group's receivers. After the first packet, the device calculates the shortest path between the receiver and source (the Shortest Path Tree, or SPT) and uses the SPT for subsequent packets from the source to the receiver. The device calculates a separate SPT for each source-receiver pair.

NOTE
It is recommended that you configure the same ports as candidate BSRs and RPs.

RP paths and SPT paths

Figure 156 shows two paths for packets from the source for group 239.255.162.1 and a receiver for the group. The source is attached to PIM Sparse device A and the recipient is attached to PIM Sparse device C. PIM Sparse device B in is the RP for this multicast group. As a result, the default path for packets from the source to the receiver is through the RP. However, the path through the RP sometimes is not the shortest path. In this case, the shortest path between the source and the receiver is over the direct link between device A and device C, which bypasses the RP (device B).

To optimize PIM traffic, the protocol contains a mechanism for calculating the Shortest Path Tree (SPT) between a given source and receiver. PIM Sparse devices can use the SPT as an alternative to using the RP for forwarding traffic from a source to a receiver. By default, the device forwards the first packet they receive from a given source to a given receiver using the RP path, but forward subsequent packets from that source to that receiver through the SPT. In Figure 156, device A forwards the first packet from group 239.255.162.1's source to the destination by sending the packet to device B, which is the RP. Device B then sends the packet to device C. For the second and all future packets that device A receives from the source for the receiver, device A forwards them directly to device C using the SPT path.

Configuring PIM Sparse

To configure a PowerConnect for PIM Sparse, perform the following tasks:

- Configure the following global parameter:
  - Enable the PIM Sparse mode of multicast routing.
- Configure the following interface parameters:
  - Configure an IP address on the interface
  - Enable PIM Sparse.
  - Identify the interface as a PIM Sparse border, if applicable.
- Configure the following PIM Sparse global parameters:
  - Identify the PowerConnect as a candidate PIM Sparse Bootstrap Router (BSR), if applicable.
  - Identify the PowerConnect as a candidate PIM Sparse Rendezvous Point (RP), if applicable.
  - Specify the IP address of the RP (if you want to statically select the RP).

NOTE
It is recommended that you configure the same PowerConnect as both the BSR and the RP.

Current limitations

The implementation of PIM Sparse in the current software release has the following limitations:

- PIM Sparse and regular PIM (dense mode) cannot be used on the same interface.
- You cannot configure or display PIM Sparse information using the Web Management Interface. (You can display some general PIM information, but not specific PIM Sparse information.)

Configuring global PIM Sparse parameters

To configure basic global PIM Sparse parameters, enter commands such as the following on each device within the PIM Sparse domain.

```
NetIron(config)# router pim
```

Syntax: [no] router pim

NOTE
You do not need to globally enable IP multicast routing when configuring PIM Sparse.

The command in this example enables IP multicast routing, and enables the PIM Sparse mode of IP multicast routing. The command does not configure the PowerConnect as a candidate PIM Sparse Bootstrap Router (BSR) and candidate Rendezvous Point (RP). You can configure a device as a PIM Sparse device without configuring the PowerConnect as a candidate BSR and RP. However, if you do configure the device as one of these, it is recommended that you configure the device as both of these. Refer to "Configuring BSRs" on page 1167.

Entering a **[no] router pim** command does the following:

- Disables PIM or DVMRP.
- Removes all configuration for PIM multicast on a PowerConnect (**router pim** level) only.

Enabling PIM Sparse for a specified VRF

To enable PIM for the VRF named "blue", use the following commands.

```
NetIron(config)# router pim vrf blue
```

Syntax: [no] router pim [vrf <*vrf-name*>]

The **vrf** parameter allows you to configure PIM (PIM-DM and PIM-SM) on the virtual routing instance (VRF) specified by the <*vrf-name*> variable. All PIM parameters available for the default router instance are configurable for a VRF-based PIM instance.

The [no] router pim vrf command behaves in the following manner:

- Entering the **router pim vrf** command to enable PIM does not require a software reload.
- Entering a **no router pim vrf** command removes all configuration for PIM multicast on the specified VRF.

Configuring PIM interface parameters

After you enable IP multicast routing and PIM Sparse at the global level, you must enable it on the individual interfaces connected to the PIM Sparse network.

To enable PIM Sparse mode on an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 2/2
NetIron(config-if-e10000-2/2)# ip address 207.95.7.1 255.255.255.0
NetIron(config-if-e10000-2/2)# ip pim-sparse
```

Syntax: [no] ip pim-sparse

The commands in this example add an IP interface to port 2/2, then enable PIM Sparse on the interface.

If the interface is on the border of the PIM Sparse domain, you also must enter the following command.

```
NetIron(config-if-e10000-2/2)# ip pim border
```

Syntax: [no] ip pim border

Configuring BSRs

In addition to the global and interface parameters described in the previous sections, you need to identify an interface on at least one device as a candidate PIM Sparse Bootstrap router (BSR) and candidate PIM Sparse Rendezvous Point (RP).

NOTE
It is possible to configure the device as only a candidate BSR or RP, but it is recommended that you configure the same interface on the same device as both a BSR and an RP.

This section describes how to configure BSRs. Refer to "Configuring RPs" on page 1167 for instructions on how to configure RPs.

To configure the device as a candidate BSR, enter commands such as the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# bsr-candidate ethernet 2/2 30 255
BSR address: 207.95.7.1, hash mask length: 30, priority: 255
```

These commands configure the PIM Sparse interface on port 2/2 as a BSR candidate, with a hash mask length of 30 and a priority of 255. The information shown in italics above is displayed by the CLI after you enter the candidate BSR configuration command.

Syntax: [no] bsr-candidate ethernet <*slot*>/<*portnum*> | loopback <*num*> | tunnel <*num*> | ve <*num*><*hash-mask-length*> [<*priority*>]

The ethernet <*slot*>/<*portnum*> | loopback <*num*> | ve <*num*> parameters specify the interface. The device will advertise the IP address of the specified interface as a candidate BSR.

- Enter ethernet <*slot*>/<*portnum*> for a physical interface (port).
- Enter ve <*num*> for a virtual interface.
- Enter loopback <*num*> for a loopback interface.
- Enter tunnel <*num*> for a GRE tunnel interface to be configured. The GRE tunnel interface is enabled under the router PIM configuration.

The <*hash-mask-length*> parameter specifies the number of bits in a group address that are significant when calculating the group-to-RP mapping. You can specify a value from 1–32.

NOTE
it is recommended that you specify 30 for IP version 4 (IPv4) networks.

The <*priority*> specifies the BSR priority. You can specify a value from 0 – 255. When the election process for BSR takes place, the candidate BSR with the highest priority becomes the BSR. The default is 0.

Configuring RPs

Enter a command such as the following to configure the device as a candidate RP.

```
NetIron(config-pim-router)# rp-candidate ethernet 2/2
```

Syntax: [**no**] **rp-candidate ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **tunnel** <*num*> | **ve** <*num*>

The **ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> parameters specify the interface. The device will advertise the IP address of the specified interface as a candidate RP.

- Enter **ethernet** <*slot*>/<*portnum*> for a physical interface (port).
- Enter **ve** <*num*> for a virtual interface.
- Enter **loopback** <*num*> for a loopback interface.
- Enter **tunnel** <*num*> for a GRE tunnel interface to be configured. The GRE tunnel interface is enabled under the router PIM configuration.

By default, this command configures the device as a candidate RP for all group numbers beginning with 224. As a result, the device is a candidate RP for all valid PIM Sparse group numbers. You can change this by adding or deleting specific address ranges. The following example narrows the group number range for which the device is a candidate RP by explicitly adding a range.

```
NetIron(config-pim-router)# rp-candidate add 224.126.0.0 16
```

Syntax: [**no**] **rp-candidate add** <*group-addr*> <*mask-bits*>

The <*group-addr*> <*mask-bits*> specifies the group address and the number of significant bits in the subnet mask. In this example, the device is a candidate RP for all groups that begin with 224.126. When you add a range, you override the default. The device then becomes a candidate RP only for the group address ranges you add.

You also can delete the configured rp-candidate group ranges by entering the following command.

```
NetIron(config-pim-router)# rp-candidate delete 224.126.22.0 24
```

Syntax: [**no**] **rp-candidate delete** <*group-addr*> <*mask-bits*>

The usage of the <*group-addr*> <*mask-bits*> parameter is the same as for the **rp-candidate add** command.

.Updating PIM-Sparse forwarding entries with new RP configuration

If you make changes to your static RP configuration, the entries in the PIM-Sparse multicast forwarding table continue to use the old RP configuration until they are aged out.

The **clear pim rp-map** command allows you to update the entries in the static multicast forwarding table immediately after making RP configuration changes. This command is meant to be used with **rp-address** command.

To update the entries in a PIM sparse static multicast forwarding table with new RP configuration, enter the following command at the privileged EXEC level of the CLI.

```
NetIron(config)# clear ip pim rp-map
```

Syntax: clear ip pim [**vrf** <*vrf-name*>] **rp-map**

Use the **vrf** option to clear the PIM sparse static multicast forwarding table for a VRF instance specified by the <*vrf-name*> variable.

Statically specifying the RP

It is recommended that you use the PIM Sparse protocol's RP election process so that a backup RP can automatically take over if the active RP router becomes unavailable. However, if you do not want the RP to be selected by the RP election process but instead you want to explicitly identify the RP by P address, use the **rp-address** command.

If you explicitly specify the RP, the device uses the specified RP for all group-to-RP mappings and overrides the set of candidate RPs supplied by the BSR.

NOTE
Specify the same IP address as the RP on all PIM Sparse devices within the PIM Sparse domain. Make sure the device is on the backbone or is otherwise well connected to the rest of the network.

To specify the IP address of the RP, enter commands such as the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# rp-address 207.95.7.1
```

Syntax: [**no**] **rp-address** <*ip-addr*>

The <*ip-addr*> parameter specifies the IP address of the RP.

The command in this example identifies the device interface at IP address 207.95.7.1 as the RP for the PIM Sparse domain. The device uses the specified RP and ignore group-to-RP mappings received from the BSR.

ACL based RP assignment

The **rp-address** command allows multiple static RP configurations. For each static RP, an ACL can be given as an option to define the multicast address ranges that the static RP permit or deny to serve.

A static RP by default serves the range of 224.0.0.0/4 if the RP is configured without an ACL name. If an ACL name is given but the ACL is not defined, the static RP is set to inactive mode and it will not cover any multicast group ranges.

The optional static RP ACL can be configured as a standard ACL or as an extended ACL. For an extended ACL, the destination filter will be used to derive the multicast group range and all other filters are ignored. The content of the ACL needs to be defined in the order of prefix length; the longest prefix must be placed at the top of the ACL definition.

If there are overlapping group ranges among the static RPs, the static RP with the longest prefix match is selected. If more than one static RP covers the exact same group range, the highest IP static RP will be used.

Configuration considerations:

- The Static RP has higher precedence over RP learnt from the BSR.
- There is a limit of 64 static RPs in the systems.

Configuring an ACL based RP assignment

To configure an ACL based RP assignment, enter commands such as the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# rp-address 130.1.1.1 acl1
```

Syntax: [**no**] **rp-address** <*ip_address*> [<*acl_name_or_id*>] [**vrf** <*vrf-name*>]

Use the **ip address** parameter to specify the IP address of the device you want to designate as an RP device.

Use the **acl name** or **id** (optional) parameter to specify the name or ID of the ACL that specifies which multicast groups use this RP.

Use the **vrf** parameter to specify a VRF instance.

Displaying the static RP

Use the **show ip pim rp-set** command to display static RP and the associated group ranges.

```
NetIron(config)# show ip pim rp-set
Static RP and associated group ranges
-------------------------------------
Static RP count: 4
130.1.1.1
   permit 238.1.1.0/24
   permit 239.1.0.0/16
   permit 235.0.0.0/8
120.1.1.1
   deny all
120.2.1.1
   deny all
124.1.1.1
   permit 224.0.0.0/4
Number of group prefixes Learnt from BSR: 0
No RP-Set present.
```

Use the **show ip pim rp-map** command to display all current multicast group addresses to RP address mapping.

```
NetIron(config)# show ip pim rp-map
Number of group-to-RP mappings: 5
      Group address     RP address
--------------------------------------
1     230.0.0.1         100.1.1.1
2     230.0.0.2         100.1.1.1
3     230.0.0.3         100.1.1.1
4     230.0.0.4         100.1.1.1
5      230.0.0.5          100.1.1.1
```

Route selection precedence for multicast

The **route-precedence** command lets you specify a precedence table that dictates how routes are selected for multicast.

Configuration considerations:

PIM must be enabled at the global level.

Configuring route precedence by specifying route types

The **route precedence** command lets you control the selection of routes based on the following route types:

- Non-default route from the mRTM
- Default route from the mRTM
- Non-default route from the uRTM
- Default route from the uRTM

Use this command to specify an option for all of the precedence levels.

Example

To specify a non-default route from the mRTM, then a non-default route from the uRTM, then a default route from the mRTM, and then a default route from the uRTM, enter commands such as the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# route-precedence mc-non-default uc-non-default
mcdefault uc-default
```

The **none** option may be used to fill up the precedence table in order to ignore certain types of routes. To use the unicast default route for multicast, enter commands such as the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# route-precedence mc-non-default mc-default
uc-non-default none
```

Syntax: [**no**] **route-precedence** [**mc-non-default** | **mc-default** | **uc-non-default** | **uc-default** | **none**]

Default value: **route-precedence mc-non-default mc-default uc-non-default uc-default**

Use the **mc-non-default** parameter to specify a multicast non-default route.

Use the **mc-default** parameter to specify a multicast default route.

Use the **uc-non-default p**arameter to specify a unicast non-default route.

Use the **uc-default** parameter to specify a unicast default route.

Use the **none** parameter to ignore certain types of routes.

The **no** form of this command removes the configuration.

Displaying the route selection

Use the **show ip pim sparse** command to display the current route selection. This example shows the default route precedence selection.

```
NetIron(config)#show ip pim sparse
Global PIM Sparse Mode Settings
  Hello interval : 30 Neighbor timeout : 105
  Bootstrap Msg interval: 60 Candidate-RP Advertisement interval: 60
  Join/Prune interval : 60 SPT Threshold : 1
  Inactivity interval : 180 SSM Enabled : No
  Hardware Drop Enabled : Yes
  Route Selection : mc-non-default mc-default uc-non-default uc-default
---------+---------------+----+---+----------------------+------+-------------+
Interface|Local          |Mode|Ver| Designated Router    |TTL   |Multicast    |
         |Address        |    |   |Address         Port  |Thresh|Boundary     |
---------+---------------+----+---+----------------------+------+-------------+
      v12 100.4.8.2        SM  V2  Itself                    1 None
      v13 100.16.8.2       SM  V2  Itself                    1 None
     v124 124.0.0.1        SM  V2  Itself                    1 None
     v125 125.0.0.1        SM  V2  Itself                    1 None
     v126 126.0.0.1        SM  V2  Itself                    1 None
     v127 127.0.0.1        SM  V2  Itself                    1 None
        l1 1.0.8.1         SM  V2   Itself                   1 None
```

In this example, the route precedence selection is multicast non-default, then unicast non-default, then multicast default, and then unicast default.

Changing the Shortest Path Tree (SPT) threshold

In a typical PIM Sparse domain, there may be two or more paths from a DR (designated router) for a multicast source to a PIM group receiver:

- **Path through the RP** – This is the path the device uses the first time it receives traffic for a PIM group. However, the path through the RP may not be the shortest path from the device to the receiver.
- **Shortest Path** – Each PIM Sparse device that is a DR for a multicast source calculates a shortest path tree (SPT) to all the PIM Sparse group receivers within the domain, with the device itself as the root of the tree. The first time a device configured as a PIM router receives a packet for a PIM receiver, the device sends the packet to the RP for the group. The device also calculates the SPT from itself to the receiver. The next time the device receives a PIM Sparse packet for the receiver, the device sends the packet toward the receiver using the shortest route, which may not pass through the RP.

By default, the device switches from the RP to the SPT after receiving the first packet for a given PIM Sparse group. The device maintains a separate counter for each PIM Sparse source-group pair.

After the device receives a packet for a given source-group pair, it starts a PIM data timer for that source-group pair. If the device does not receive another packet for the source-group pair before the timer expires, it reverts to using the RP for the next packet received for the source-group pair. In accordance with the PIM Sparse RFC recommendation, the timer is 210 seconds and is not configurable. The counter is reset to zero each time the device receives a packet for the source-group pair.

You can change the number of packets that the device sends using the RP before switching to using the SPT by entering commands such as the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# spt-threshold 1000
```

Syntax: [**no**] **spt-threshold infinity** | <*num*>

The **infinity** | <*num*> parameter specifies the number of packets. If you specify **infinity**, the device sends packets using the RP indefinitely and does not switch over to the SPT. If you enter a specific number of packets, the device does not switch over to using the SPT until it has sent the number of packets you specify using the RP.

Configuring PIM-SM (*,g) forwarding

By default, the device supports only (s,g) hardware routing. This is adequate in network topologies where there a limited number of multicast flows, and most SPT and RPF paths diverge from the PIM Last Hop (PIM LH) from the PIM First Hop (PIM FH).

If however, the RPF path from a PIM LH to the PIM RP and the source is the same, the intermediate devices between RP and LH can be optimized to aggregate multicast flows destined to the same group address to use a single (*,g) CAM entry, instead of consuming an (s,g) hardware entry for each flow. This reduces CAM usage as well as the number of IPCs between the interface and managment modules to manage the individual (s,g) states.

For example, where a service provider is provisioning multicast service, with only the route of PIM RP being visible to customers, and routes to the sources are all default, the (*,g) hardware entry can help optimize system resources by keeping the traffic on the shared tree.

NOTE
It is recommended to configure the **spt-threshold infinity** command beginning from the PIM LH router, then to the intermediate PIM routers and finally to the PIM RP.

Configuration details

To enable this feature, you must explicitly configure the **spt-threshold infinity** command on all multicast nodes, as shown in the following example.

```
NetIron(config)# router pim
NetIron(config-pim-router)# spt-threshold infinity
```

Syntax: [**no**] **spt-threshold infinity**

This configuration option forces PIM-SM to distribute multicast traffic on the share tree only. PIM devices from RP to LH maintain only (*,g) states, regardless of the source location in the network topology. PIM LH will not initiate (s,g) SPT switchover, so there are no (s,g) joins generated from LH. Devices in the share tree, eg, devices downstream from RP, create a (*,g) hardware forwarding state to switch multicast flows for the group.

Changing the PIM Join and Prune message interval

By default, the device sends PIM Sparse Join or Prune messages every 60 seconds. These messages inform other PIM Sparse devices about clients who want to become receivers (Join) or stop being receivers (Prune) for PIM Sparse groups.

NOTE
Use the same Join or Prune message interval on all the PIM Sparse devices in the PIM Sparse domain. If the devices do not all use the same timer interval, the performance of PIM Sparse can be adversely affected.

To change the Join or Prune interval, enter commands such as the following.

```
NetIron(config)# router pim
NetIron(config-pim-router)# message-interval 30
```

Syntax: [**no**] **message-interval** <*num*>

The <*num*> parameter specifies the number of seconds and can from 1 – 65535. The default is 60.

Multicast Outgoing Interface (OIF) list optimization

Each multicast route entry maintains a list of outgoing interfaces (OIF List) to which an incoming multicast data packet matching the route is replicated. In hardware-forwarded route entries, these OIF lists are stored inside the hardware in replication tables which are limited in size. In many deployment scenarios, more than one multicast route can have identical OIF lists and can optimize usage of the replication table entries by sharing them across multiple multicast routes. Multicast OIF list optimization keeps track of all the OIF lists in the system. It manages the hardware replication resources optimally, in real time, by dynamically assigning or re-assigning resources to multicast route entries to suit their current OIF list requirements, while maximizing resource sharing.

Displaying PIM Sparse configuration information and statistics

You can display the following PIM Sparse information:

- Basic PIM Sparse configuration information
- Group information
- BSR information
- Candidate RP information
- RP-to-group mappings
- RP information for a PIM Sparse group
- RP set list
- PIM neighbor information
- The PIM flow cache
- The PIM multicast cache
- PIM traffic statistics
- PIM counter statistics

Displaying basic PIM Sparse configuration information

To display PIM Sparse configuration information, enter the following command at any CLI level.

```
NetIron# show ip pim sparse
Global PIM Sparse Mode Settings
  Hello interval        : 30          Neighbor timeout                  : 105
  Bootstrap Msg interval: 60          Candidate-RP Advertisement interval: 60
  Join/Prune interval   : 60          SPT Threshold                     : 1
  Inactivity interval   : 180
  SSM Enabled: Yes
  SSM Group Range: 226.0.0.0/8
---------+---------------+----+---+----------------------+------
Interface|Local           |Mode|Ver|  Designated Router   |TTL
         |Address         |    |   |Address          Port |Thresh
---------+---------------+----+---+----------------------+------
     e6/3 10.2.0.10         SM  V2  Itself                     1
     e6/4 10.3.0.10         SM  V2  Itself                     1
      v30 10.10.10.10       SM  V2  Itself                     1
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **sparse**

The **vrf** option allows you to display PIM sparse configuration information for the VRF instance identified by the <*vrf-name*> variable.

This example shows the PIM Sparse configuration information on PIM Sparse device A in Figure 156.

Table 181 shows the information displayed by the **show ip pim sparse** command..

TABLE 181 Output of the **show ip pim sparse** command

This field...	Displays...
Global PIM Sparse mode settings	
Hello interval	How frequently the device sends PIM Sparse hello messages to its PIM Sparse neighbors. This field also shows the number of seconds between hello messages. PIM Sparse devices use hello messages to discover each another.
Neighbor timeout	How many seconds the device waits for a hello message from a neighbor before determining that the neighbor is no longer present and removing cached PIM Sparse forwarding entries for the neighbor.
Bootstrap Msg interval	How frequently the BSR configured on the device sends the RP set to the RPs within the PIM Sparse domain. The RP set is a list of candidate RPs and their group prefixes. A candidate RP group prefix indicates the range of PIM Sparse group numbers for which it can be an RP. **NOTE:** This field contains a value only if an interface on the device is elected to be the BSR. Otherwise, the field is blank.
Candidate-RP Advertisement interval	How frequently the candidate PR configured on the device sends candidate RP advertisement messages to the BSR. **NOTE:** This field contains a value only if an interface on the device is configured as a candidate RP. Otherwise, the field is blank.
Join or Prune interval	How frequently the device sends PIM Sparse Join or Prune messages for the multicast groups it is forwarding. This field also shows the number of seconds between Join or Prune messages. The device sends Join or Prune messages on behalf of multicast receivers who want to join or leave a PIM Sparse group. When forwarding packets from PIM Sparse sources, the device sends the packets only on the interfaces on which it has received join requests in Join or Prune messages for the source group. You can change the Join or Prune interval. Refer to "Changing the PIM Join and Prune message interval" on page 1173.
SPT Threshold	The number of packets the device sends using the path through the RP before switching to using the SPT path.
Inactivity Interval	
SSM Enabled	If yes, source-specific multicast is configured globally on this device.
SSM Group Range	The SSM range of IP multicast addresses. By default this range will be 232/8 as assigned by the Internet Assigned Numbers Authority (IANA) for use with SSM. Other values can be configured.
PIM Sparse interface information **NOTE:** You also can display IP multicast interface information using the **show ip pim interface** command. However, this command lists all IP multicast interfaces, including regular PIM (dense mode) and DVMRP interfaces. The **show ip pim sparse** command lists only the PIM Sparse interfaces.	
Interface	The type of interface and the interface number. The interface type can be one of the following: • Ethernet • VE The number is either a port number (and slot number if applicable) or the virtual interface (VE) number.
TTL Threshold	Following the TTL threshold value, the interface state is listed. The interface state can be one of the following: • Disabled • Enabled

TABLE 181 Output of the **show ip pim sparse** command (Continued)

This field...	Displays...
Local Address	Indicates the IP address configured on the port or virtual interface.
Mode	
Version	
Designated Router	

Displaying a list of multicast groups

To display PIM group information, enter the following command at any CLI level.

```
NetIron# show ip pim group

Total number of Groups: 2
Index 1         Group 239.255.162.1       Ports e3/11
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **group**

The **vrf** option allows you to display PIM group information for the VRF instance identified by the <*vrf-name*> variable.

Table 182 describes the output from this command.

TABLE 182 Output from the **show ip pim vrf group** command

This field...	Displays...
Total number of Groups	Lists the total number of IP multicast groups the device is forwarding. **NOTE:** This list can include groups that are not PIM Sparse groups. If interfaces on the device are configured for regular PIM (dense mode) or DVMRP, these groups are listed too.
Index	The index number of the table entry in the display.
Group	The multicast group address
Ports	The device ports connected to the receivers of the groups.

Displaying BSR information

To display BSR information, enter the following command at any CLI level.

```
NetIron# show ip pim bsr

PIMv2 Bootstrap information

This system is the elected Bootstrap Router (BSR)
  BSR address: 207.95.7.1
  Uptime: 00:33:52, BSR priority: 5, Hash mask length: 32
  Next bootstrap message in 00:00:20

Next Candidate-RP-advertisement in 00:00:10
  RP: 207.95.7.1
    group prefixes:
    224.0.0.0 / 4
Candidate-RP-advertisement period: 60
```

This example shows information displayed on a device that has been elected as the BSR. The next example shows information displayed on a device that is not the BSR. Notice that some fields shown in the example above do not appear in the example below.

```
NetIron# show ip pim bsr

PIMv2 Bootstrap information
 BSR address  = 207.95.7.1
 BSR priority = 5
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **bsr**

The **vrf** option allows you to display BSR information for the VRF instance identified by the <*vrf-name*> variable.

Table 183 describes the output from this command.

TABLE 183 Output from the **show ip pim bsr** command

This field...	Displays...
BSR address	The IP address of the interface configured as the PIM Sparse Bootstrap Router (BSR).
Uptime	The amount of time the BSR has been running. **NOTE:** This field appears only if this device is the BSR.
BSR priority	The priority assigned to the interface for use during the BSR election process. During BSR election, the priorities of the candidate BSRs are compared and the interface with the highest BSR priority becomes the BSR.
Hash mask length	The number of significant bits in the IP multicast group comparison mask. This mask determines the IP multicast group numbers for which the device can be a BSR. The default is 32 bits, which allows the device to be a BSR for any valid IP multicast group number. **NOTE:** This field appears only if this device is a candidate BSR.
Next bootstrap message in	Indicates how many seconds will pass before the BSR sends the next bootstrap message. **NOTE:** This field appears only if this device is the BSR.
Next Candidate-RP-advertisement message in	Indicates how many seconds will pass before the BSR sends the next candidate PR advertisement message. **NOTE:** This field appears only if this device is a candidate BSR.
RP	Indicates the IP address of the Rendezvous Point (RP). **NOTE:** This field appears only if this device is a candidate BSR.
group prefixes	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP. **NOTE:** This field appears only if this device is a candidate BSR.
Candidate-RP-advertisement period	Indicates how frequently the BSR sends candidate RP advertisement messages. **NOTE:** This field appears only if this device is a candidate BSR.

Displaying candidate RP information

To display candidate RP information, enter the following command at any CLI level.

```
NetIron# show ip pim rp-candidate

Next Candidate-RP-advertisement in 00:00:10
  RP: 207.95.7.1
    group prefixes:
    224.0.0.0 / 4

  Candidate-RP-advertisement period: 60
```

This example show information displayed on a device that is a candidate RP. The next example shows the message displayed on a device that is not a candidate RP.

```
NetIron# show ip pim rp-candidate
```

This system is not a Candidate-RP.

Syntax: show ip pim rp-candidate <*vrf-name*>

This command displays candidate RP information for the VRF instance identified by the <*vrf-name*> variable.

Table 184 describes the output from this command.

TABLE 184 Output from the **show ip pim rp-candidate** command

This field...	Displays...
Candidate-RP-advertisement in	Indicates how many seconds will pass before the BSR sends the next RP message. **NOTE:** This field appears only if this device is a candidate RP.
RP	Indicates the IP address of the Rendezvous Point (RP). **NOTE:** This field appears only if this device is a candidate RP.
group prefixes	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP. **NOTE:** This field appears only if this device is a candidate RP.
Candidate-RP-advertisement period	Indicates how frequently the BSR sends candidate RP advertisement messages. **NOTE:** This field appears only if this device is a candidate RP.

Displaying RP-to-group mappings

To display RP-to-group-mappings, enter the following command at any CLI level.

```
NetIron# show ip pim rp-map
Number of group-to-RP mappings: 6

Group address        RP address
------------------------------
1 239.255.163.1  99.99.99.5
2 239.255.163.2  99.99.99.5
3 239.255.163.3  99.99.99.5
4 239.255.162.1  99.99.99.5
5 239.255.162.2  43.43.43.1
6 239.255.162.3  99.99.99.5
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **rp-map**

The **vrf** option allows you to display candidate RP-to-group mappings for the VRF instance identified by the <*vrf-name*> variable.

This display shows the following information.

TABLE 185 Output of the **show ip pim rp-map** command

This field...	Displays...
Group address	Indicates the PIM Sparse multicast group address using the listed RP.
RP address	Indicates the IP address of the Rendezvous Point (RP) for the listed PIM Sparse group.

Displaying RP Information for a PIM Sparse group

To display RP information for a PIM Sparse group, enter the following command at any CLI level.

```
NetIron# show ip pim rp-hash 239.255.162.1

  RP: 207.95.7.1, v2
    Info source: 207.95.7.1, via bootstrap
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **rp-hash** <*group-addr*>

The **vrf** option allows you to display RP information for the VRF instance identified by the <*vrf-name*> variable.

The <*group-addr*> parameter is the address of a PIM Sparse IP multicast group.

Table 186 describes the output from this command.

TABLE 186 Output from the **show ip pim** command

This field...	Displays...
RP	Indicates the IP address of the Rendezvous Point (RP) for the specified PIM Sparse group, followed by the port or virtual interface through which this device learned the identity of the RP.
Info source	Indicates the IP address on which the RP information was received, followed by the IP address through which this device learned the identity of the RP.

Displaying the RP set list

To display the RP set list, enter the following command at any CLI level.

```
NetIron# show ip pim rp-set
Group address Static-RP-address Override
-------------------------------------------------
Access-List 44 99.99.99.5 On
Number of group prefixes Learnt from BSR: 1
Group prefix = 239.255.162.0/24 # RPs expected: 1
# RPs received: 1
RP 1: 43.43.43.1 priority=0 age=0
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **rp-set**

The **vrf** option allows you to display the RP set list for the VRF instance identified by the <*vrf-name*> variable.

Table 187 describes the output from this command.

TABLE 187 Output from the **show ip pim vrf rp-set** command

This field...	Displays...
Number of group prefixes	The number of PIM Sparse group prefixes for which the RP is responsible.
Group prefix	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP.
RPs expected or received	Indicates how many RPs were expected and received in the latest bootstrap message.
RP <*num*>	Indicates the RP number. If there are multiple RPs in the PIM Sparse domain, a line of information for each RP is listed, in ascending numerical order.
priority	The RP priority of the candidate RP. During the election process, the candidate RP with the highest priority is elected as the RP.
age	The age (in seconds) of this RP-set. **NOTE:** If this device is not a BSR, this field contains zero. Only the BSR ages the RP-set.

Displaying multicast neighbor information

To display information about PIM neighbors, enter the following command at any CLI level.

```
NetIron# show ip pim nbr

Port   Phy_Port Neighbor      Holdtime  Age    UpTime  VRF          Priority
                              sec       sec    sec
e1/8   e1/3     207.95.8.10   105       30     900     default-VRF   1
e2/4   e2/4     65.31.11.1    105       30     60      default-VRF   10
p3/1   e3/      165.31.11.1   105       30     300     default-VRF   N/A
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **nbr**

The **vrf** option allows you to display information about the PIM neighbors for the VRF instance identified by the <*vrf-name*> variable.

Table 188 describes the output from this command.

TABLE 188 Output from the **show ip pim vrf nbr** command

This field...	Displays...
Port	The interface through which the device is connected to the neighbor.
Neighbor	The IP interface of the PIM neighbor.
Holdtime sec	Indicates how many seconds the neighbor wants this device to hold the entry for this neighbor in memory. The neighbor sends the Hold Time in Hello packets: • If the device receives a new Hello packet before the Hold Time received in the previous packet expires, the device updates its table entry for the neighbor. • If the device does not receive a new Hello packet from the neighbor before the Hold time expires, the device assumes the neighbor is no longer available and removes the entry for the neighbor.
Age sec	The number of seconds since the device received the last hello message from the neighbor.
UpTime sec	The number of seconds the PIM neighbor has been up. This timer starts when the device receives the first Hello messages from the neighbor.

TABLE 188 Output from the **show ip pim vrf nbr** command (Continued)

This field...	Displays...
VRF	The VRF in which the interface is configured. This can be a VRF that the port was assigned to or the default VRF of the device.
Priority	The DR priority that is used in the DR election process. This can be a configured value or the default value of 1.

Displaying the PIM multicast cache

To display the PIM multicast cache, enter the following command at any CLI level.

```
NetIron# show ip pim mcache
Total 6 entries
1 (10.161.32.200, 237.0.0.1) in v87 (tag e3/1), cnt=0
   Sparse Mode, RPT=0 SPT=1 Reg=0
   upstream neighbor=10.10.8.45
   num_oifs = 1 v2
   L3 (HW) 1: e4/24(VL2702)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: 0416 l2vidx: none
2 (*, 237.0.0.1) RP10.161.2.1 in v93, cnt=0
   Sparse Mode, RPT=1 SPT=0 Reg=0
   upstream neighbor=10.10.8.33
   num_oifs = 1 v2
   L3 (SW) 1: e4/24(VL2702)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: none l2vidx: none
3 (*, 239.255.255.250) RP10.159.2.2 in v87, cnt=0
   Sparse Mode, RPT=1 SPT=0 Reg=0
   upstream neighbor=10.10.8.45
   num_oifs = 1 v2
   L3 (SW) 1: e4/23(VL2702)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: none l2vidx: none
4 (137.80.133.220, 224.225.0.3) in v16 (tag e1/3)
   upstream neighbor=172.17.42.2
   L3 (HW) 2: e1/4(VL15), e1/3(VL11)
   L2 (HW) 1: TR(e1/5,e1/6)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=1 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: 0409 l2vidx: 040f
5 (137.80.200.124, 224.225.0.4) in v200 (tag e1/3)
   Source is directly connected
   L3 (HW) 1: e1/4(VL15)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=1 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: 0410 l2vidx: none
6 (137.80.134.232, 224.225.0.5) in v16 (tag e1/3)
   upstream neighbor=172.17.42.2
   L3 (HW) 2: e1/3(VL11), e1/4(VL200)
   L2 (HW) 1: TR(e1/5,e1/5)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=1 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: 0402 l2vidx: 0408
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **mcache**

The **vrf** option allows you to display the PIM multicast cache for the VRF instance identified by the <*vrf-name*> variable.

Displaying the PIM multicast cache for MVID

To display the PIM multicast cache for a specified mvid, enter the following command at any CLI level.

```
NetIron# show ip pim mcache mvid 1
Total 8 entries
1    (1.1.1.100, 239.1.1.1) in e3/1 (e3/1), cnt=0
     Sparse Mode, RPT=0 SPT=1 Reg=0 L2Reg=0 RegSupp=0 LSrc=1 LRcv=1
     Source is directly connected. RP 100.2.1.1
     num_oifs = 2
     L3 (HW) 2: e3/15(VL10 VL20)
     Flags (0x004680a1)
        sm=1 ssm=0 fast=1 slow=0 leaf=0 prun=0 frag=0 tag=1 needRte=0 msdp_adv=1
     age=0 fid: 0x8008  mvid: 0x0001
2    (1.1.1.100, 239.1.1.3) in e3/1 (e3/1), cnt=0
     Sparse Mode, RPT=0 SPT=1 Reg=0 L2Reg=0 RegSupp=0 LSrc=1 LRcv=1
     Source is directly connected. RP 100.2.1.1
     num_oifs = 2
     L3 (HW) 2: e3/15(VL10 VL20)
     Flags (0x004680a1)
        sm=1 ssm=0 fast=1 slow=0 leaf=0 prun=0 frag=0 tag=1 needRte=0 msdp_adv=1
     age=0 fid: 0x8008  mvid: 0x0001

Total number of mcache entries 2
```

Syntax: show ip pim mcache mvid <*mvid*>

The <*mvid*> variable allows you to display an entry that matches a specified mvid.

Displaying the PIM multicast cache for FID

To display the PIM multicast cache for a specified fid, enter the following command at any CLI level

```
NetIron# show ip pim mcache fid 8009
Total 8 entries
1    (1.1.1.100, 239.1.1.2) in e3/1 (e3/1), cnt=0
     Sparse Mode, RPT=0 SPT=1 Reg=0 L2Reg=0 RegSupp=0 LSrc=1 LRcv=1
     Source is directly connected. RP 100.2.1.1
     num_oifs = 3
     L3 (HW) 3: e3/15(VL10 VL20 VL30)
     Flags (0x004680a1)
        sm=1 ssm=0 fast=1 slow=0 leaf=0 prun=0 frag=0 tag=1 needRte=0 msdp_adv=1
     age=0 fid: 0x8009  mvid: 0x0000
2    (1.1.1.100, 239.1.1.3) in e3/1 (e3/1), cnt=0
     Sparse Mode, RPT=0 SPT=1 Reg=0 L2Reg=0 RegSupp=0 LSrc=1 LRcv=1
     Source is directly connected. RP 100.2.1.1
     num_oifs = 2
     L3 (HW) 2: e3/15(VL10 VL20)
     Flags (0x004680a1)
        sm=1 ssm=0 fast=1 slow=0 leaf=0 prun=0 frag=0 tag=1 needRte=0 msdp_adv=1
     age=0 fid: 0x8009  mvid: 0x0001

Total number of mcache entries 2
```

Syntax: show ip pim mcache fid <*fid-id*>

The <*fid-id*> variable allows you to display an entry that matches a specified fid.

Clearing the PIM forwarding cache

You can clear the PIM forwarding cache using the following command.

```
NetIron# clear ip pim cache
```

Syntax: clear ip pim [**vrf** <*vrf-name*>] **cache**

Use the **vrf** option to clear the PIM forwarding cache for a VRF instance specified by the <*vrf-name*> variable.

Displaying PIM traffic statistics

To display PIM traffic statistics, enter the following command at any CLI level.

```
NetIron# show ip pim traffic

Port    Hello           J/P          Register       RegStop          Assert
     [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]
e3/8   19     19      32     0       0      0       37     0       0      0
v1     18     19      0      20      0      0       0      0       0      0
v2     0      19      0      0       0      16      0      0       0      0

Total 37      57     32      0       0      0       0      0       0      0
IGMP Statistics:
  Total Recv/Xmit 85/110
  Total Discard/chksum  0/0
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **traffic**

The **vrf** option allows you to display the PIM traffic statistics for the VRF instance identified by the <*vrf-name*> variable.

NOTE
If you have configured interfaces for standard PIM (dense mode) on the device, statistics for these interfaces are listed first by the display.

Table 189 describes the output from this command.

TABLE 189 Output from the **show ip pim vrf traffic** command

This field...	Displays...
Port	The port or virtual interface on which the PIM interface is configured.
Hello	The number of PIM Hello messages sent or received on the interface.
J or P	The number of Join or Prune messages sent or received on the interface. **NOTE:** Unlike PIM dense, PIM Sparse uses the same messages for Joins and Prunes.
Register	The number of Register messages sent or received on the interface.
RegStop	The number of Register Stop messages sent or received on the interface.
Assert	The number of Assert messages sent or received on the interface.

TABLE 189 Output from the **show ip pim vrf traffic** command (Continued)

This field...	Displays...
Total Recv or Xmit	The total number of IGMP messages sent and received by the device.
Total Discard or chksum	The total number of IGMP messages discarded, including a separate counter for those that failed the checksum comparison.

Clearing the PIM message counters

You can clear the PIM message counters using the following command.

```
NetIron# clear ip pim traffic
```

Syntax: clear ip pim [**vrf** <*vrf-name*>] **traffic**

Use the **vrf** option to clear the PIM message counters for a VRF instance specified by the <*vrf-name*> variable.

Displaying PIM RPF

The **show ip pim rfp** command displays what PIM sees as the reverse path to the source as shown in the following. While there may be multiple routes back to the source, the one displayed by this command is the one that PIM thinks is best.

```
NetIron# show ip pim vrf eng rpf 130.50.11.10
Source 130.50.11.10 directly connected on e4/1
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **rpf** <*ip-address*>

The <*ip-address*> variable specifies the source address for RPF check.

The **vrf** option to display what PIM sees as the reverse path to the source for a VRF instance specified by the <*vrf-name*> variable.

Displaying PIM counters

You can display the number of default-vlan-id changes that have occurred since the applicable VRF was created, and how many times a tagged port was placed in a VLAN since the applicable VRF was created as shown.

```
NetIron(config)# show ip pim vrf eng counter
Event Callback:
   DFTVlanChange :          2               VlanPort:          13
```

Syntax: show ip pim [**vrf** <*vrf-name*>] **counter**

Table 190 describes the output from this command.

TABLE 190 Output from the **show ip pim vrf counter** command

This field...	Displays...
DFTVlanChange	The number of default-vlan-id changes that have occurred since the applicable VRF was created.
VlanPort	The number of times that a tagged port was placed in a VLAN since the applicable VRF was created.

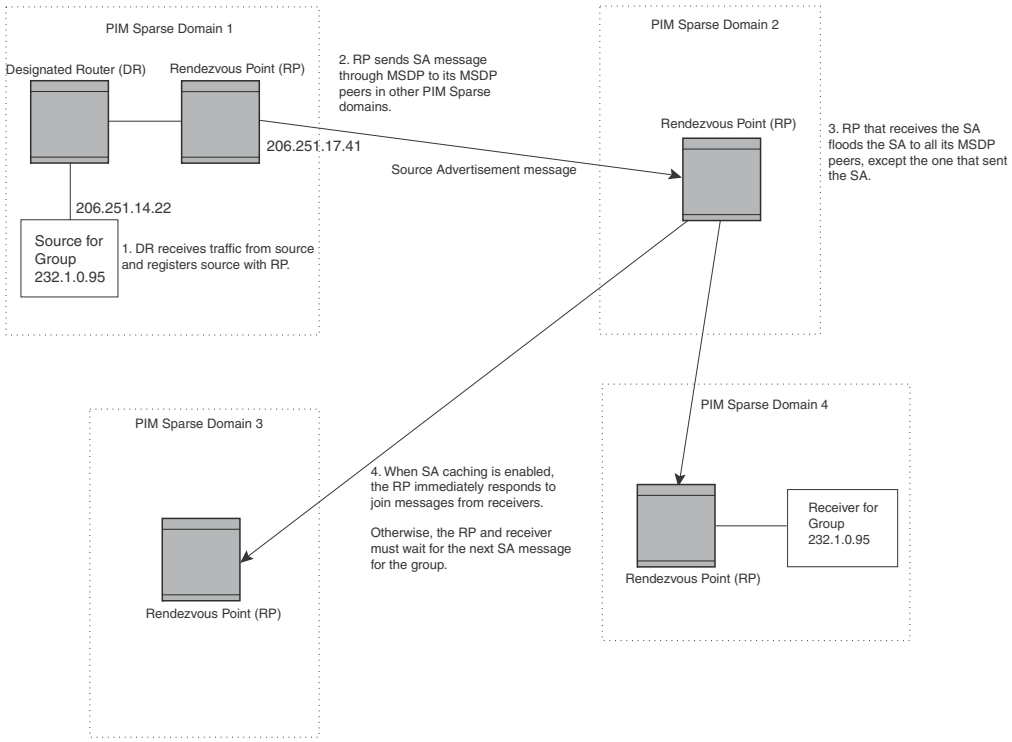
NOTE
Since VLANs are not VRF-aware, any changes to default-vlan or tagged port moves is counted by all VRFs in existence at the time, including the default VRF.

Configuring Multicast Source Discovery Protocol (MSDP)

The Multicast Source Discovery Protocol (MSDP) is used by Protocol Independent Multicast (PIM) Sparse devices to exchange source information across PIM Sparse domains. Devices running MSDP can discover PIM Sparse sources in other PIM Sparse domains.

Figure 157 shows an example of some PIM Sparse domains. For simplicity, this example shows one Designated Router (DR), one group source, and one receiver for the group. Only one PIM Sparse device within each domain needs to run MSDP.

FIGURE 157 PIM Sparse domains joined by MSDP devices



In this example, the source for PIM Sparse multicast group 232.0.1.95 is in PIM Sparse domain 1. The source sends a packet for the group to its directly attached DR. The DR sends a Group Advertisement message for the group to the RP for the domain. The RP is configured for MSDP, which enables the RP to exchange source information with other PIM Sparse domains by communicating with RPs in other domains that are running MSDP.

The RP sends the source information to each peer through a Source Active message. The message contains the IP address of the source, the group address to which the source is sending, and the IP address of the RP.

In this example, the Source Active message contains the following information:

- Source address: 206.251.14.22
- Group address: 232.1.0.95
- RP address: 206.251.17.41

Figure 157 shows only one peer for the MSDP device (which is also the RP here) in domain 1, so the Source Active message goes to only that peer. When an MSDP device has multiple peers, it sends a Source Active message to each of those peers. Each peer sends the Source Advertisement to other MSDP peers. The RP that receives the Source Active message also sends a Join message to the source if the RP that received the message has receivers for the group and source.

Peer Reverse Path Forwarding (RPF) flooding

When the MSDP device (also the RP) in domain 2 receives the Source Active message from the peer in domain 1, the MSDP device in domain 2 forwards the message to all other peers. This propagation process is sometimes called "peer Reverse Path Forwarding (RPF) flooding". In Figure 157, the MSDP device floods the Source Active message it receives from the peer in domain 1 to peers in domains 3 and 4.

The MSDP device in domain 2 does not forward the Source Active back to the peer in domain 1, because that is the peer from which the device received the message. An MSDP device never sends a Source Active message back to the peer that sent it. The peer that sent the message is sometimes called the "RPF peer". The MSDP device uses the unicast routing table for its Exterior Gateway Protocol (EGP) to identify the RPF peer by looking for the route entry that is the next hop toward the source. Often, the EGP protocol is Border Gateway Protocol (BGP) version 4.

NOTE
MSDP depends on BGP and MBGP for inter-domain operations.

The MSDP routers in domains 3 and 4 also forward the Source Active message to all peers except the ones that sent them the message. Figure 157 does not show additional peers.

Source Active caching

When an MSDP device that is also an RP and source receives a Source Active message, the RP and source checks the PIM Sparse multicast group table for receivers for the group. If the DR has a receiver for the group being advertised in the Source Active message, the RP sends a Join message towards that source.

In Figure 157, if the MSDP device and RP in domain 4 has a table entry for the receiver, the RP sends a Join message on behalf of the receiver back through the RPF tree to the source, in this case the source in domain 1.

Source Active caching is enabled in MSDP on Dell devices. The RP caches the Source Active messages it receives even if the RP does not have a receiver for the group. Once a receiver arrives, the RP can then send a Join to the cached source immediately.

The size of the cache used to store MSDP Source Active messages is 32K.

Configuring MSDP

To configure MSDP, perform the following tasks:

- Enable MSDP.
- Configure the MSDP peers.

NOTE
The PIM Sparse Rendezvous Point (RP) is also an MSDP peer.

NOTE
Devices that run MSDP usually also run BGP. The source address used by the MSDP device is normally configured to be the same source address used by BGP.

Enabling MSDP

To enable MSDP, enter the following command.

```
NetIron(config)# router msdp
```

Syntax: [no] router msdp

Enabling MSDP for a specified VRF

The **vrf** parameter allows you to configure MSDP on the virtual routing instance (VRF) specified by the <*vrf-name*> variable. All MSDP parameters available for the default router instance are configurable for a VRF-based MSDP instance.

To enable MSDP for the VRF named “blue”, enter the following commands.

```
NetIron(config)# router msdp vrf blue
NetIron(config-msdp-router-vrf-blue)
```

Syntax: [no] router msdp [vrf <*vrf-name*>]

The **vrf** parameter allows you to configure MSDP on the virtual routing instance (VRF) specified by the <*vrf-name*> variable.

Entering a **no router msdp vrf** command removes the MSDP configuration from the specified VRF only.

Assigning a Route Distinguisher to a VRF

Each instance of a VRF must be assigned a unique Route Distinguisher (RD). The RD is prepended on any address being routed or advertised. The RD can be defined as either ASN-relative or IP address-relative. Since the RD is unique to an instance of a VRF, it allows the same IP address to be used in different VPNs without creating any conflict.

To assign a Route Distinguisher (RD) for a VRF based on the AS number 3 and the arbitrary identification number 6, enter the following command.

```
NetIron(config)# router msdp vrf blue
NetIron(config-msdp-router-vrf-blue)# rd 3:6
NetIron(config-msdp-router-vrf-blue) exit
```

Syntax: [no] rd <*route_distinguisher*>

The <*route_distinguisher*> variable specifies a route distinguisher for a VRF that gives a unique identity to a route associated with the VRF. The RD is prepended on the address being advertised. The RD allows the same IP address to be used in different VPNs without creating any conflicts. It can also be used with the **route-target** command to constrain distribution of routes to or from a VPN. The <*route_distinguisher*> parameter can be either ASN-relative or IP address-relative.

ASN-relative – Composed of the local ASN number followed by a “:” and a unique arbitrary number. For example: 3:6

IP address-relative – Composed of the local IP address followed by a ":" and a unique arbitrary number.

Configuring MSDP peers

To configure an MSDP peer, enter a command such as the following at the MSDP configuration level.

```
NetIron(config-msdp-router)# msdp-peer 205.216.162.1
```

To configure an MSDP peer on a VRF, enter the following commands at the MSDP VRF configuration level.

```
NetIron(config)# router msdp vrf blue
NetIron(config-msdp-router-vrf-blue)# msdp-peer 205.216.162.1
```

Syntax: [no] **msdp-peer** <*ip-addr*> [**connect-source loopback** <*num*>]

The <*ip-addr*> parameter specifies the IP address of the neighbor.

The **connect-source loopback** <*num*> parameter specifies the loopback interface you want to use as the source for sessions with the neighbor and must be reachable within the VRF.

NOTE
It is strongly recommended that you use the connect-source loopback <*num*> parameter when issuing the **msdp-peer** command. If you do not use this parameter, the device uses the IP address of the outgoing interface. You should also make sure the IP address of the connect-source loopback is the source IP address used by the PIM-RP, and the BGP device.

The commands in the following example add an MSDP neighbor and specify a loopback interface as the source interface for sessions with the neighbor. By default, the device uses the subnet address configured on the physical interface where you configure the neighbor as the source address for sessions with the neighbor.

```
NetIron(config)# interface loopback 1
NetIron(config-lbif-1)# ip address 9.9.9.9/32
NetIron(config)# router msdp
NetIron(config-msdp-router)# msdp-peer 2.2.2.99 connect-source loopback 1
```

Disabling an MSDP peer

To disable an MSDP peer, enter the following command at the configure MSDP router level.

```
NetIron(config-msdp-router)# msdp-peer 205.216.162.1 shutdown
```

To disable the MSDP VRF peer named "blue", enter the following commands.

```
NetIron(config)# router msdp vrf blue
NetIron(config-msdp-router-vrf-blue)# no msdp-peer 205.216.162.1
```

Syntax: [no] **msdp-peer** <*ip-addr*> **shutdown**

The <*ip-addr*> parameter specifies the IP address of the MSDP peer that you want to disable.

Designating the interface IP address as the RP IP address

When an RP receives a Source Active message, it checks its PIM Sparse multicast group table for receivers for the group. If a receiver exists the RP sends a Join to the source.

By default, the IP address included in the RP address field of the SA message is the IP address of the originating RP. An SA message can use the IP address of any interface on the originating RP. (The interface is usually a loopback interface.)

To designate an interface IP address to be the IP address of the RP, enter commands such as the following.

```
NetIron(config)# interface loopback 2
NetIron(config-lbif-2)# ip address 2.2.1.99/32
NetIron(config)# router msdp
NetIron(config-msdp-router)# originator-id loopback 2
NetIron(config-msdp-router)# exit
```

To specify VRF information, enter the following commands at the MSDP VRF configuration level.

```
NetIron(config)# interface loopback 2
NetIron(config-lbif-2)# ip address 2.2.1.99/32
NetIron(config)# router msdp vrf blue
NetIron(config-msdp-router-vrf blue)# originator-id loopback 2
NetIron(config-msdp-router-vrf blue)# exit
```

Syntax: [no] originator-id <*type*> <*number*>

The originator-id parameter instructs MSDP to use the specified interface IP address as the IP address of the RP in an SA message. This address must be the address of the interface used to connect the RP to the source. The default address used is the RP IP address.

The <*type*> parameter indicates the type of interface used by the RP. Ethernet, loopback and virtual routing interfaces (ve) can be used.

The <*number*> parameter specifies the interface number (for example: loopback number, port number or virtual routing interface number.)

Filtering MSDP source-group pairs

You can filter individual source-group pairs in MSDP Source-Active messages:

- sa-filter in – Filters source-group pairs received in Source-Active messages from an MSDP neighbor.
- sa-filter originate – Filters self-originated source-group pairs in outbound Source-Active messages sent to an MSDP neighbor
- sa-filter out – Filters self-originated and forwarded source-group pairs in outbound Source-Active messages sent to an MSDP neighbor

Filtering incoming and outgoing Source-Active messages

The following example configures filters for incoming Source-Active messages from three MSDP neighbors:

- For peer 2.2.2.99, all source-group pairs in Source-Active messages from the neighbor are filtered (dropped).
- For peer 2.2.2.97, all source-group pairs except those with source address matching 10.x.x.x and group address of 235.10.10.1 are permitted.
- For peer 2.2.2.96, all source-group pairs except those associated with RP 2.2.42.3 are permitted.

To configure filters for incoming Source-Active messages, enter commands at the MSDP VRF configuration level.

To configure filters for outbound Source-Active messages, enter the optional out keyword.

Example

The following commands configure extended ACLs. The ACLs will be used in route maps, which will be used by the Source-Active filters.

```
NetIron(config)# access-list 123 permit ip 10.0.0.0 0.255.255.255 host
235.10.10.1
NetIron(config)# access-list 124 permit ip host 2.2.42.3 any
NetIron(config)# access-list 125 permit ip any any
```

The following commands configure the route maps.

```
NetIron(config)# route-map msdp_map deny 1
NetIron(config-routemap msdp_map)# match ip address 123
NetIron(config-routemap msdp_map)# exit
NetIron(config)# route-map msdp_map permit 2
NetIron(config-routemap msdp_map)# match ip address 125
NetIron(config-routemap msdp_map)# exit
NetIron(config)# route-map msdp2_map permit 1
NetIron(config-routemap msdp2_map)# match ip address 125
NetIron(config-routemap msdp2_map)# exit
NetIron(config)# route-map msdp2_rp_map deny 1
NetIron(config-routemap msdp2_rp_map)# match ip route-source 124
NetIron(config-routemap msdp2_rp_map)# exit
NetIron(config)# route-map msdp2_rp_map permit 2
NetIron(config-routemap msdp2_rp_map)# match ip route-source 125
NetIron(config-routemap msdp2_rp_map)# exit
```

To specify VRF information, enter the following commands at the MSDP VRF configuration level.

```
NetIron(config)# router msdp vrf blue
NetIron(config-msdp-router-vrf blue)# sa-filter in 2.2.2.99
NetIron(config-msdp-router-vrf blue)# sa-filter in 2.2.2.97 route-map msdp_map
NetIron(config-msdp-router-vrf blue)# sa-filter in 2.2.2.96 route-map msdp2_map
rp-route-map msdp2_rp_map
```

The **sa-filter** commands configure the following filters:

- **sa-filter in 2.2.2.99** – This command drops all source-group pairs received from neighbor 2.2.2.99.

  NOTE
  The default action is to deny all source-group pairs from the specified neighbor. If you want to permit some pairs, use route maps.

- **sa-filter in 2.2.2.97 route-map msdp_map** – This command drops source-group pairs received from neighbor 2.2.2.97 if the pairs have source addresses matching 10.x.x.x and group address 235.10.10.1.
- **sa-filter in 2.2.2.96 route-map msdp2_map rp-route-map msdp2_rp_map** – This command accepts all source-group pairs except those associated with RP 2.2.42.3.

Syntax: [**no**] **sa-filter in** | **originate** | **out** <*ip-addr*> [**route-map** <*map-tag*>] [**rp-route-map** <*rp-map-tag*>]

Selecting the **in** option applies the filter to incoming Source-Active messages.

Selecting the **originate** option applies the filter to self-originated outbound Source-Active messages.

Selecting the **out** option applies the filter to self-originated and forwarded outbound Source-Active messages.

The <*ip-addr*> parameter specifies the IP address of the MSDP neighbor. The filters apply to Source-Active messages received from or sent to this neighbor.

The **route-map** <*map-tag*> parameter specifies a route map. The device applies the filter to source-group pairs that match the route map. Use the **match ip address** <*acl-id*> command in the route map to specify an extended ACL that contains the source addresses.

The **rp-route-map** <*rp-map-tag*> parameter specifies a route map to use for filtering based on Rendezvous Point (RP) address. Use this parameter if you want to filter Source-Active messages based on their originating RP. Use the **match ip route-source** <*acl-id*> command in the route map to specify an extended ACL that contains the RP address.

NOTE
The default filter action is deny. If you want to permit some source-group pairs, use a route map.

Filtering advertised Source-Active messages

The following example configures the device to advertise all source-group pairs except the ones that have source address 10.x.x.x.

Example

The following commands configure extended ACLs to be used in the route map definition.

```
NetIron(config)# access-list 123 permit ip 10.0.0.0 0.255.255.255 any
NetIron(config)# access-list 125 permit ip any any
```

The following commands use the above ACLs to configure a route map which denies source-group with source address 10.x.x.x and any group address, while permitting everything else.

```
NetIron(config)# route-map msdp_map deny 1
NetIron(config-routemap msdp_map)# match ip address 123
NetIron(config-routemap msdp_map)# exit
NetIron(config)# route-map msdp_map permit 2
NetIron(config-routemap msdp_map)# match ip address 125
NetIron(config-routemap msdp_map)# exit
```

The following commands configure the Source-Active filter.

```
NetIron(config)# router msdp
NetIron(config-msdp-router)# sa-filter originate route-map msdp_map
```

To specify VRF information, enter the following commands at the MSDP VRF configuration level.

```
NetIron(config)# router msdp vrf blue
NetIron(config-msdp-router-vrf blue)# sa-filter originate route-map msdp_map
```

Syntax: [**no**] **sa-filter originate** [**route-map** <*map-tag*>]

The **route-map** <*map-tag*> parameter specifies a route map. The router applies the filter to source-group pairs that match the route map. Use the **match ip address** <*acl-id*> command in the route map to specify an extended ACL that contains the source and group addresses.

NOTE
The default filter action is deny. If you want to permit some source-group pairs, use a route map. A permit action in the route map allows the device to advertise the matching source-group pairs. A deny action in the route map drops the source-group pairs from advertisements.

Displaying MSDP information

You can display the following MSDP information:

- **Summary information** – the IP addresses of the peers, the state of the device MSDP session with each peer, and statistics for keepalive, source active, and notification messages sent to and received from each of the peers
- **VRF Information** – Summary information for a specific VRF
- **Peer information** – the IP address of the peer, along with detailed MSDP and TCP statistics
- **Source Active cache entries** – the source active messages cached by the router.

Displaying summary information

To display summary MSDP information, enter the CLI command.

```
NetIron# show ip msdp vrf blue summary

MSDP Peer Status Summary
 KA: Keepalive SA:Source-Active NOT: Notification
 Peer Address     State          KA          SA          NOT
                              In     Out   In    Out   In   Out
 206.251.17.30    ESTABLISH   3      3     0     640   0     0
 206.251.17.41    ESTABLISH   0      3     651   0     0     0
```

Syntax: show ip msdp summary

Table 191 describes the output from this command.

TABLE 191 MSDP summary information

This field...	Displays...
Peer address	The IP address of the peer interface with the device
State	The state of the MSDP device connection with the peer. The state can be one of the following: • CONNECTING – The session is in the active open state. • ESTABLISHED – The MSDP session is fully up. • INACTIVE – The session is idle. • LISTENING – The session is in the passive open state.
KA In	The number of MSDP keepalive messages the MSDP device has received from the peer
KA Out	The number of MSDP keepalive messages the MSDP device has sent to the peer
SA In	The number of source active messages the MSDP device has received from the peer
SA Out	The number of source active messages the MSDP device has sent to the peer
NOT In	The number of notification messages the MSDP router has received from the peer
NOT Out	The number of notification messages the MSDP router has sent to the peer

Displaying peer information

To display MSDP peer information, enter the following command.

```
NetIron# show ip msdp vrf blue peer

        Total number of MSDP Peers: 2

        IP Address           State
1       206.251.17.30        ESTABLISHED
        Keep Alive Time  Hold Time
        60               90

                             Message Sent         Message Received
        Keep Alive           2                    3
        Notifications        0                    0
        Source-Active        0                    640
        Last Connection Reset Reason:Reason Unknown
        Notification Message Error Code Received:Unspecified
        Notification Message Error SubCode Received:Not Applicable
        Notification Message Error Code Transmitted:Unspecified
        Notification Message Error SubCode Transmitted:Not Applicable
        TCP Connection state: ESTABLISHED
          Local host:  206.251.17.29, Local  Port: 8270
          Remote host: 206.251.17.30, Remote Port: 639
          ISentSeq:      16927  SendNext:     685654  TotUnAck:          0
          SendWnd:       16384  TotSent:      668727  ReTrans:           1
          IRcvSeq:    45252428  RcvNext:    45252438  RcvWnd:        16384
          TotalRcv:         10  RcvQue:            0  SendQue:           0
```

Syntax: show ip msdp peer

Table 192 describes the output from this command.

TABLE 192 MSDP peer information

This field...	Displays...
Total number of MSDP peers	The number of MSDP peers configured on the device
IP Address	The IP address of the peer's interface with the device
State	The state of the MSDP device connection with the peer. The state can be one of the following: • CONNECTING – The session is in the active open state. • ESTABLISHED – The MSDP session is fully up. • INACTIVE – The session is idle. • LISTENING – The session is in the passive open state.
Keep Alive Time	The keepalive time, which specifies how often this MSDP device sends keep alive messages to the neighbor. The keep alive time is 60 seconds and is not configurable.
Hold Time	The hold time, which specifies how many seconds the MSDP device will wait for a KEEPALIVE or UPDATE message from an MSDP neighbor before deciding that the neighbor is dead. The hold time is 90 seconds and is not configurable.
Keep Alive Message Sent	The number of keepalive messages the MSDP device has sent to the peer.

TABLE 192 MSDP peer information (Continued)

This field...	Displays...
Keep Alive Message Received	The number of keepalive messages the MSDP device has received from the peer.
Notifications Sent	The number of notification messages the MSDP device has sent to the peer.
Notifications Received	The number of notification messages the MSDP device has received from the peer.
Source-Active Sent	The number of source active messages the MSDP device has sent to the peer.
Source-Active Received	The number of source active messages the MSDP device has received from the peer.
Last Connection Reset Reason	The reason the previous session with this neighbor ended.
Notification Message Error Code Received	The MSDP device has received a notification message from the neighbor that contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages: • 1 – Message Header Error • 2 – SA-Request Error • 3 – SA-Message or SA-Response Error • 4 – Hold Timer Expired • 5 – Finite State Machine Error • 6 – Notification • 7 – Cease For information about these errors, refer to section 17 in the Internet draft describing MSDP, “draft-ietf-msdp-spec”.
Notification Message Error SubCode Received	See above.
Notification Message Error Code Transmitted	The error message corresponding to the error code in the NOTIFICATION message this MSDP router sent to the neighbor. See the description for the Notification Message Error Code Received field for a list of possible codes.
Notification Message Error SubCode Transmitted	See above.
TCP Statistics	

TABLE 192 MSDP peer information (Continued)

This field...	Displays...
TCP connection state	The state of the connection with the neighbor. Can be one of the following: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (includes an acknowledgment of the connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of the connection termination request. • CLOSED – There is no connection state.
Local host	The IP address of the MSDP device interface with the peer.
Local port	The TCP port the MSDP router is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port number of the peer end of the connection.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the MSDP device that have not been acknowledged by the neighbor.
SendWnd	The size of the send window.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers the MSDP device retransmitted because they were not acknowledged.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
RcvWnd	The size of the receive window.
TotalRcv	The number of sequence numbers received from the neighbor.
RcvQue	The number of sequence numbers in the receive queue.
SendQue	The number of sequence numbers in the send queue.

Displaying Source Active cache information

To display the Source Actives in the MSDP cache, use the following command.

```
NetIron# show ip msdp vrf blue sa-cache
Total of 10 SA cache entries
Index  RP address       (Source, Group)                    Orig Peer        Age
1      2.2.2.2          (192.6.1.10, 227.1.1.1)            192.1.1.2        0
2      2.2.2.2          (192.6.1.10, 227.1.1.2)            192.1.1.2        0
3      2.2.2.2          (192.6.1.10, 227.1.1.3)            192.1.1.2        0
4      2.2.2.2          (192.6.1.10, 227.1.1.4)            192.1.1.2        0
5      2.2.2.2          (192.6.1.10, 227.1.1.5)            192.1.1.2        0
6      2.2.2.2          (192.6.1.10, 227.1.1.6)            192.1.1.2        0
7      2.2.2.2          (192.6.1.10, 227.1.1.7)            192.1.1.2        0
8      2.2.2.2          (192.6.1.10, 227.1.1.8)            192.1.1.2        0
9      2.2.2.2          (192.6.1.10, 227.1.1.9)            192.1.1.2        0
10     2.2.2.2          (192.6.1.10, 227.1.1.10)           192.1.1.2        0
```

Syntax: show ip msdp sa-cache

Table 193 describes the output from this command.

TABLE 193 MSDP source active cache

This field...	Displays...
Total	The number of entries the cache currently contains.
Index	The cache entry number.
RP	The RP through which receivers can access the group traffic from the source
SourceAddr	The IP address of the multicast source.
GroupAddr	The IP multicast group to which the source is sending information.
Orig Peer	The peer from which this source-active entry was received.
Age	The number of seconds the entry has been in the cache

Clearing MSDP information

You can clear the following MSDP information:

- Peer information
- Source active cache
- MSDP statistics

Clearing peer information

To clear MSDP peer information, enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# clear ip msdp peer 205.216.162.1
```

Syntax: clear ip msdp peer <*ip-addr*>

The command in this example clears the MSDP peer connection with MSDP router 205.216.162.1. The CLI displays a message to indicate when the connection has been successfully closed. To clear all the peers, omit the <*ip-addr*> variable from the command.

Clearing peer information on a VRF

To clear the MSDP VRF peers, enter the following command at the MSDP VRF configuration level.

```
NetIron#clear ip msdp vrf blue peer 207.207.162.5
```

Clearing the source active cache

To clear the source active cache, enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# clear ip msdp sa-cache
```

Syntax: clear ip msdp sa-cache <*ip-addr*>

The command in this example clears all the cache entries. Use the <*ip-addr*> variable to clear only the entries matching either a source or a group.

Clearing the source active cache for a VRF

To clear the MSDP VRF source active cache by entering the following command at the MSDP VRF configuration level.

```
NetIron#clear ip msdp sa-cache vrf blue
```

Syntax: clear ip msdp sa-cache [**vrf**<*vrf-name*>] [<*ip-addr*>]

Clearing MSDP statistics

To clear MSDP statistics, enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# clear ip msdp statistics
```

Syntax: clear ip msdp statistics <*ip-addr*>

The command in this example clears statistics for all the peers. To clear statistics for only a specific peer, enter the IP address of the peer.

Clearing MSDP VRF statistics

To clear the MSDP VRF statistics by entering the following command.

```
NetIron# clear ip msdp vrf blue statistics
```

Syntax: clear ip msdp statistics [**vrf**<*vrf-name*>] [<*ip-addr*>]

The command in this example clears statistics for all the peers. To clear statistics for only a specific peer, enter the IP address of the peer.

The command in this example clears all statistics for all the peers in the VRF “blue.”.

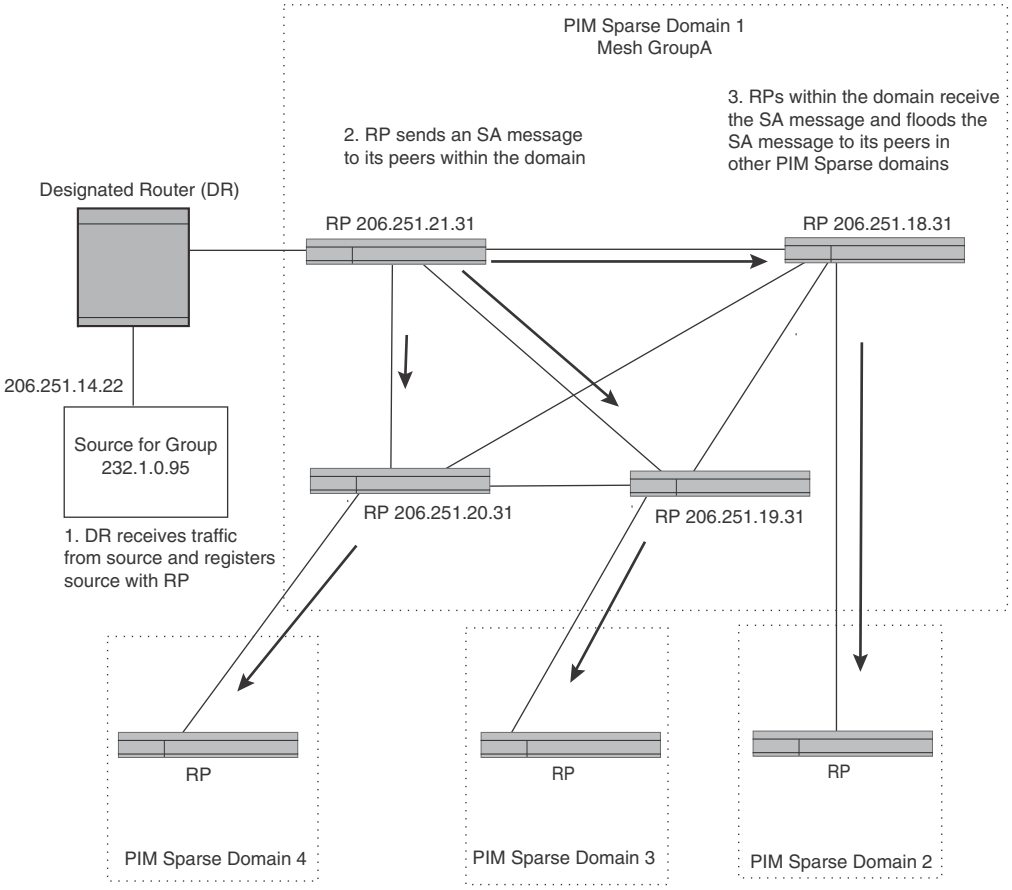
Configuring MSDP mesh groups

A PIM Sparse domain can have several RPs that are connected to each other to form an MSDP mesh group. To qualify as a mesh group, the RPs have to be fully meshed; that is, each RP must be connected to all peer RPs in a domain. (Refer to Figure 158.)

A mesh group reduces the forwarding of SA messages within a domain. Instead of having every RP in a domain forward SA messages to all the RPs within that domain, only one RP forwards the SA message. Since an MSDP mesh group is fully meshed, peers do not forward SA messages received in a domain from one member to any member of the group. The RP that originated the SA or the first RP in a domain that receives the SA message is the only one that forwards the message to the members of a mesh group. An RP can forward an SA message to any MSDP router as long as that peer is farther away from the originating RP than the current MSDP router.

Figure 158 shows an example of an MSDP mesh group. In a PIM-SM mesh group the RPs are configured to be peers of each other. They can also be peers of RPs in other domains.

FIGURE 158 Example of MSDP mesh group



PIM Sparse Domain 1 in Figure 158 contains a mesh group with four RPs. When the first RP, for example, RP 206.251.21.31 originates or receives an SA message from a peer in another domain, it sends the SA message to its peers within the mesh group. However, the peers do not send the message back to the originator RP or to each other. The RPs then send the SA message farther away to their peers in other domains.The process continues until all RPs within the network receive the SA message.

Configuring MSDP mesh group

To configure an MSDP mesh group, enter commands such as the following on each device that will be included in the mesh group.

```
NetIron(config)# router msdp
NetIron(config-msdp-router)# msdp-peer 206.251.18.31 connect-source loopback 2
NetIron(config-msdp-router)# msdp-peer 206.251.19.31 connect-source loopback 2
NetIron(config-msdp-router)# msdp-peer 206.251.20.31 connect-source loopback 2
NetIron(config-msdp-router)# mesh-group GroupA 206.251.18.31
NetIron(config-msdp-router)# mesh-group GroupA 206.251.19.31
NetIron(config-msdp-router)# mesh-group GroupA 206.251.20.31
NetIron(config-msdp-router)# exit
```

Syntax: [no] mesh-group <*group-name*> <*peer-address*>

The sample configuration above reflects the configuration in Figure 158. On RP 206.251.21.31 you specify its peers within the same domain (206.251.18.31, 206.251.19.31, and 206.251.20.31).

You first configure the MSDP peers using the **msdp-peer** command to assign their IP addresses and the loopback interfaces.

Next, place the MSDP peers within a domain into a mesh group. Use the **mesh-group** command. There are no default mesh groups.

The **group-name** parameter identifies the mesh group. Enter up to 31 characters for group-name. You can have up to 4 mesh groups within a multicast network. Each mesh group can include up to 15 peers.

The **peer-address** parameter specifies the IP address of the MSDP peer that is being placed in the mesh group.

NOTE
On each of the device that will be part of the mesh group, there must be a mesh group definition for all the peers in the mesh-group.

A maximum of 15 MSDP peers can be configured per mesh group.

MSDP Anycast RP

MSDP Anycast RP is a method of providing intra-domain redundancy and load-balancing between multiple Rendezvous Points (RP) in a Protocol Independent Multicast Sparse mode (PIM-SM) network. It is accomplished by configuring all RPs within a domain with the same anycast RP address which is typically a loopback IP address. Multicast Source Discovery Protocol (MSDP) is used between all of the RPs in a mesh configuration to keep all RPs in sync regarding the active sources.

PIM-SM routers are configured to register (statically or dynamically) with the RP using the same anycast RP address. Since multiple RPs have the same anycast address, an Interior Gateway Protocol (IGP) such as OSPF routes the PIM-SM router to the RP with the best route. If the PIM-SM routers are distributed evenly throughout the domain, the loads on RPs within the domain will be distributed. If the RP with the best route goes out of service, the PIM-SM router's IGP changes the route to the closest operating RP that has the same anycast address.

This configuration works because MSDP is configured between all of the RPs in the domain. Consequently, all of the RPs share information about active sources.

This feature uses functionality that is already available on the PowerConnect Router but re-purposes it to provide the benefits desired as described in RFC 3446.

Configuring MSDP Anycast RP

To configure MSDP Anycast RP, you must perform the following tasks:

- Configure a loopback interface with the anycast RP address on each of the RPs within the domain and enable PIM-SM on these interfaces.
- Ensure that the anycast RP address is leaked into the IGP domain. This is typically done by enabling the IGP on the loopback interface (in passive mode) or redistributing the connected loopback IP address into the IGP.

NOTE
The anycast RP address *must* not be the IGP router-id.

- Enable PIM-SM on all interfaces on which multicast routing is desired.
- Enable an IGP on each of the loopback interfaces and physical interfaces configured for PIM-SM.
- Configure loopback interfaces with unique IP addresses on each of the RPs for MSDP peering. This loopback interface is also used as the MSDP originator-id.
- The non-RP PIM-SM routers may be configured to use the anycast RP address statically or dynamically (by the PIMv2 bootstrap mechanism).

Example

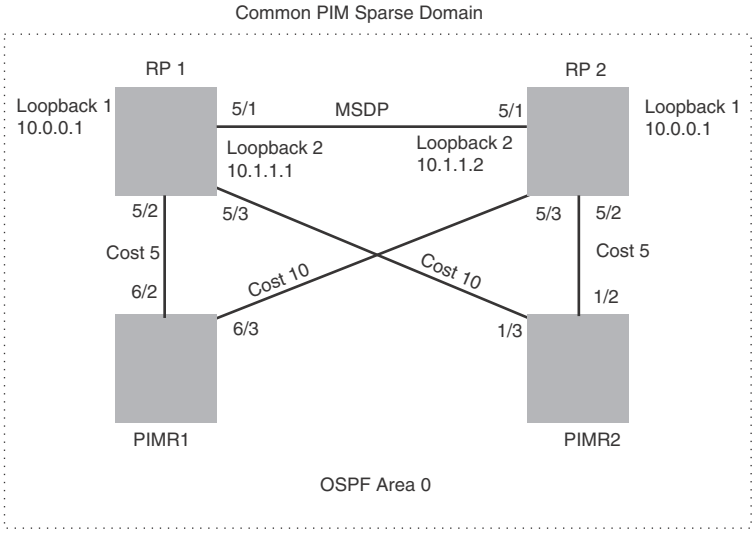
The example shown in Figure 159 is a simple MSDP Anycast-enabled network with two RPs and two PIM-SM routers. Loopback 1 in RP 1 and RP 2 have the same IP address. Loopback 2 in RP1 and Loopback 2 in RP2 have different IP addresses and are configured as MSDP peering IP addresses in a mesh configuration.

In the PIM configuration for PIM-SM routers PIMR1 and PIMR2 the RP address is configured to be the anycast RP address that was configured on the Loopback 1 interfaces on RP1 and RP2. OSPF is configured as the IGP for the network and all of the devices are in OSPF area 0.

Since PIMR1 has a lower cost path to RP1 and PIMR2 has a lower cost path to RP2 they will register with the respective RPs when both are up and running. This shares the load between the two RPs. If one of the RPs fails, the higher-cost path to the IP address of Loopback 1 on the RPs is used to route to the still-active RP.

The configuration examples demonstrate the commands required to enable this application.

FIGURE 159 Example of a MDSP Anycast RP network



RP 1 configuration

The following commands provide the configuration for the RP 1 router in Figure 159.

```
RP1(config)#router ospf
RP1(config-ospf-router)# area 0
RP1(config-ospf-router)# exit
RP1(config)# interface loopback 1
RP1(config-lbif-1)# ip ospf area 0
RP1(config-lbif-1)# ip ospf passive
RP1(config-lbif-1)# ip address 10.0.0.1/32
RP1(config-lbif-1)# ip pim-sparse
RP1(config-lbif-1)# exit
RP1(config)# interface loopback 2
RP1(config-lbif-2)# ip ospf area 0
RP1(config-lbif-2)# ip ospf passive
RP1(config-lbif-2)# ip address 10.1.1.1/32
RP1(config-lbif-2)# exit
RP1(config)# interface ethernet 5/1
RP1(config-if-e1000-5/1)# ip ospf area 0
RP1(config-if-e1000-5/1)# ip address 192.1.1.1/24
RP1(config-if-e1000-5/1)# ip pim-sparse
RP1(config)# interface ethernet 5/2
RP1(config-if-e1000-5/2)# ip ospf area 0
RP1(config-if-e1000-5/2)# ip ospf cost 5
RP1(config-if-e1000-5/2)# ip address 192.2.1.1/24
RP1(config-if-e1000-5/2)# ip pim-sparse
RP1(config)# interface ethernet 5/3
RP1(config-if-e1000-5/3)# ip ospf area 0
RP1(config-if-e1000-5/3)# ip ospf cost 10
RP1(config-if-e1000-5/3)# ip address 192.3.1.1/24
RP1(config-if-e1000-5/3)# ip pim-sparse
RP1(config-if-e1000-5/3)# exit
RP1(config)# router pim
RP1(config-pim-router)# rp-candidate loopback 1
RP1(config-pim-router)# exit
RP1(config)# router msdp
RP1(config-msdp-router)# msdp-peer 10.1.1.2 connect-source loopback 2
RP1(config-msdp-router)# originator-id loopback 2
```

RP 2 configuration

The following commands provide the configuration for the RP 2 router in Figure 159.

```
RP2(config)#router ospf
RP2(config-ospf-router)# area 0
RP2(config-ospf-router)# exit
RP2(config)# interface loopback 1
RP2(config-lbif-1)# ip ospf area 0
RP2(config-lbif-1)# ip ospf passive
RP2(config-lbif-1)# ip address 10.0.0.1/32
RP2(config-lbif-1)# ip pim-sparse
RP2(config-lbif-1)# exit
RP2(config)# interface loopback 2
RP2(config-lbif-2)# ip ospf area 0
RP2(config-lbif-2)# ip ospf passive
RP2(config-lbif-2)# ip address 10.1.1.2/32
RP2(config-lbif-2)# exit
RP2(config)# interface ethernet 5/1
RP2(config-if-e1000-5/1)# ip ospf area 0
RP2(config-if-e1000-5/1)# ip address 192.1.1.2/24
RP2(config-if-e1000-5/1)# ip pim-sparse
RP2(config)# interface ethernet 5/2
RP2(config-if-e1000-5/2)# ip ospf area 0
RP2(config-if-e1000-5/2)# ip ospf cost 5
RP2(config-if-e1000-5/2)# ip address 192.5.2.1/24
RP2(config-if-e1000-5/2)# ip pim-sparse
RP2(config)# interface ethernet 5/3
RP2(config-if-e1000-5/3)# ip ospf area 0
RP2(config-if-e1000-5/3)# ip ospf cost 10
RP2(config-if-e1000-5/3)# ip address 192.6.1.2/24
RP2(config-if-e1000-5/3)# ip pim-sparse
RP2(config-if-e1000-5/3)# exit
RP2(config)# router pim
RP2(config-pim-router)# rp-candidate loopback 1
RP2(config-pim-router)# exit
RP2(config)# router msdp
RP2(config-msdp-router)# msdp-peer 10.1.1.1 connect-source loopback 2
RP2(config-msdp-router)# originator-id loopback 2
```

PIMR1 configuration

The following commands provide the configuration for the PIMR1 router in Figure 159.

```
PIMR1(config)#router ospf
PIMR1(config-ospf-router)# area 0
PIMR1(config-ospf-router)# exit
PIMR1(config)# interface ethernet 6/2
PIMR1(config-if-e1000-6/2)# ip ospf area 0
PIMR1(config-if-e1000-6/2)# ip ospf cost 5
PIMR1(config-if-e1000-6/2)# ip address 192.2.1.2/24
PIMR1(config-if-e1000-6/2)# ip pim-sparse
PIMR1(config)# interface ethernet 6/3
PIMR1(config-if-e1000-6/3)# ip ospf area 0
PIMR1(config-if-e1000-6/3)# ip ospf cost 10
PIMR1(config-if-e1000-6/3)# ip address 192.6.1.1/24
PIMR1(config-if-e1000-6/3)# ip pim-sparse
PIMR1(config-if-e1000-6/3)# exit
```

```
PIMR1(config)# router pim
PIMR1(config-pim-router)# rp-address 10.0.0.1
PIMR1(config-pim-router)# exit
```

PIMR2 configuration

The following commands provide the configuration for the PIMR2 router in Figure 159.

```
PIMR2(config)#router ospf
PIMR2(config-ospf-router)# area 0
PIMR2(config-ospf-router)# exit
PIMR2(config)# interface ethernet 1/2
PIMR2(config-if-e1000-1/2)# ip ospf area 0
PIMR2(config-if-e1000-1/2)# ip ospf cost 5
PIMR2(config-if-e1000-1/2)# ip address 192.5.2.2/24
PIMR2(config-if-e1000-1/2)# ip pim-sparse
PIMR2(config)# interface ethernet 1/3
PIMR2(config-if-e1000-1/3)# ip ospf area 0
PIMR2(config-if-e1000-1/3)# ip ospf cost 10
PIMR2(config-if-e1000-1/3)# ip address 192.3.1.2/24
PIMR2(config-if-e1000-1/3)# ip pim-sparse
PIMR2(config-if-e1000-1/3)# exit
PIMR2(config)# router pim
PIMR2(config-pim-router)# rp-address 10.0.0.1
PIMR2(config-pim-router)# exit
```

PIM Anycast RP

PIM Anycast RP is a method of providing load balancing and fast convergence to PIM RPs in an IPv4 multicast domain. The RP address of the Anycast RP is a shared address used among multiple PIM routers, known as PIM RP. The PIM RP routers create an Anycast RP set. Each router in the Anycast RP set is configured using two IP addresses; a shared RP address in their loopback address and a separate, unique ip address. The loopback address must be reachable by all PIM routers in the multicast domain. The separate, unique ip address is configured to establish static peering with other PIM routers and communication with the peers.

When the source is activated in a PIM Anycast RP domain, the PIM First Hop (FH) will register the source to the closet PIM RP. The PIM RP follows the same MSDP Anycast RP operation by decapsulating the packet and creating the (s,g) state. If there are external peers in the Anycast RP set, the router will re-encapsulate the packet with the local peering address as the source address of the encapsulation. The router will unicast the packet to all Anycast RP peers. The re-encapsulation of the data register packet to Anycast RP peers ensures source state distribution to all RPs in a multicast domain.

Configuring PIM Anycast RP

A new PIM CLI is introduced for PIM Anycast RP under the router pim sub mode. The PIM CLI specifies mapping of the RP and the Anycast RP peers.

To configure PIM Anycast RP, enter the following command.

```
NetIron(config)#router pim
NetIron(config-pim-router)#rp-address 100.1.1.1
NetIron(config-pim-router)#anycast-rp 100.1.1.1 my-anycast-rp-set-acl
```

Syntax: [no] anycast-rp <*rp-address*> <*anycast-rp-set-acl*>

The <*rp address*> parameter specifies a shared RP address used among multiple PIM routers.

The <*anycast-rp-set-acl*> parameter specifies a host based simple acl used to specifies the address of the Anycast RP set, including a local address.

The following example is a configuration of PIM Anycast RP 100.1.1.1.The example avoids using loopback 1 interface when configuring PIM Anycast RP because the loopback 1 address could be used as a router-id. A PIM First Hop router will register the source with the closest RP. The first RP that receives the register will re-encapsulate the register to all other Anycast RP peers. Please refer to Figure 160 as described in the configuration of PIM Anycast RP 100.1.1.1.

```
NetIron(config)#interface loopback 2
NetIron(config-lbif-2)#ip address 100.1.1.1/24
NetIron(config-lbif-2)#ip pim-sparse
NetIron(config-lbif-2)#interface loopback 3
NetIron(config-lbif-3)#ip address 1.1.1.1/24
NetIron(config-lbif-3)#ip pim-sparse
NetIron(config-lbif-3)#router pim
NetIron(config-pim-router)#rp-address 100.1.1.1
NetIron(config-pim-router)#anycast-rp 100.1.1.1 my-anycast-rp-set
NetIron(config-pim-router)#ip access-list standard my-anycast-rp-set
NetIron(config-std-nacl)#permit host 1.1.1.1
NetIron(config-std-nacl)#permit host 2.2.2.2
NetIron(config-std-nacl)#permit host 3.3.3.3
```

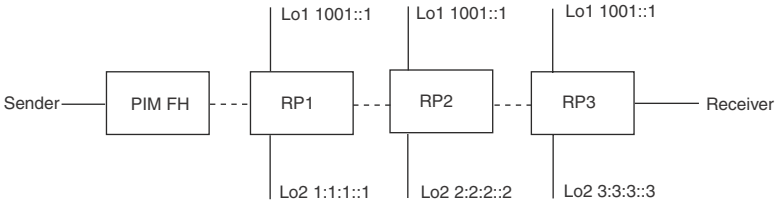
The RP shared address 100.1.1.1 is used in the PIM domain. IP addresses 1.1.1.1, 2.2.2.2, and 3.3.3.3 are listed in the ACL that forms the self inclusive Anycast RP set. Multiple anycast-rp instances can be configured on a system; each peer with the same or different Anycast RP set.

NOTE
The PIM Anycast CLI applies to only PIM routers running RP. All deny statements in the anycast_rp_set acl are ignored.

Example

The example shown in Figure 160 is a PIM Anycast-enabled network with 3 RPs, 1 PIM-FH router connecting to its active source and local receiver. Loopback 2 in RP1, RP2, and RP3 have the same IP addresses 100.1.1.1. Loopback 3 in RP1, RP2, and RP3 each have separate IP addresses configured to communicate with their peers in the Anycast RP set.

FIGURE 160 Example of a PIM Anycast RP network



Displaying information for a PIM Anycast RP interface

To display information for a PIM Anycast RP interface, enter the following command.

```
NetIron(config)#show ip pim anycast-rp
Number of Anycast RP: 1
Anycast RP: 100.1.1.1
  ACL ID: 200
  ACL Name: my-anycast-rp-set
  ACL Filter: SET
  Peer List:
    1.1.1.1
    2.2.2.2
    3.3.3.3
```

Syntax: show ip pim <*anycast-rp*>

The following table describes the parameters of the **show ip pim anycast-rp** command:

TABLE 194 Display of show ip pim-anycast-rp

This field...	Displays...
Number of Anycast RP:	The Number of Anycast RP specifies the number of Anycast RP sets in the multicast domain.
Anycast RP:	The Anycast RP address specifies a shared RP address used among multiple PIM routers.
ACL ID:	The ACL ID specifies the ACL ID assigned.
ACL Name	The ACL Name specifies the name of the Anycast RP set.
ACL Filter	The ACL Filter specifies the ACL filter state SET or UNSET.
Peer List	The Peer List specifies host addresses that are permitted in the Anycast RP set.

NOTE
MSDP and Anycast RP do not interoperate. If transitioning from MSDP to Anycast RP or vice versa, all RPs in the network must be configured for the same method of RP peering; either Anycast RP or MSDP.

DVMRP overview

The PowerConnect provides multicast routing with the Distance Vector Multicast Routing Protocol (DVMRP) routing protocol. DVMRP uses IGMP to manage the IP multicast groups.

DVMRP is a broadcast and pruning multicast protocol that delivers IP multicast datagrams to its intended receivers. The receiver registers the interested groups using IGMP. DVMRP builds a multicast delivery tree with the sender forming the root. Initially, multicast datagrams are delivered to all nodes on the tree. Those leaves that do not have any group members send ***prune messages*** to the upstream router, noting the absence of a group. The upstream router maintains a prune state for this group for the given sender. A prune state is aged out after a given configurable interval, allowing multicasts to resume.

DVMRP employs ***reverse path forwarding*** and ***pruning*** to keep source specific multicast delivery trees with the minimum number of branches required to reach all group members. DVMRP builds a multicast tree for each source and destination host group.

Initiating DVMRP multicasts on a network

Once DVMRP is enabled on each router, a network user can begin a video conference multicast from the server on R1. ***Multicast Delivery Trees*** are initially formed by source-originated multicast packets that are propagated to downstream interfaces as seen in Figure 161. When a multicast packet is received on a DVMRP-capable router interface, the interface checks its DVMRP routing table to determine whether the interface that received the message provides the shortest path back to the source. If the interface does provide the shortest path, the interface forwards the multicast packet to adjacent peer DVMRP routers, except for the router interface that originated the packet. Otherwise, the interface discards the multicast packet and sends a prune message back upstream. This process is known as ***reverse path forwarding***.

In Figure 161, the root node (R1) is forwarding multicast packets for group 229.225.0.2 that it receives from the server to its downstream nodes, R2, R3, and R4. Router R4 is an intermediate router with R5 and R6 as its downstream routers. Because R5 and R6 have no downstream interfaces, they are leaf nodes.

The receivers in this example are those workstations that are resident on routers R2, R3, and R6.

Pruning a multicast tree

After the multicast tree is constructed, ***pruning*** of the tree will occur after IP multicast packets begin to traverse the tree.

As multicast packets reach leaf networks (subnets with no downstream interfaces), the local IGMP database checks for the recently arrived IP multicast packet address. If the local database does not contain the address (the address has not been learned), the router prunes (removes) the address from the multicast tree and no longer receives multicasts until the prune age expires.

In Figure 162, Router 5 is a leaf node with no group members in its local database. Consequently, Router 5 sends a prune message to its upstream router. This router will not receive any further multicast traffic until the prune age interval expires.

FIGURE 161 Downstream broadcast of IP multicast packets from source host

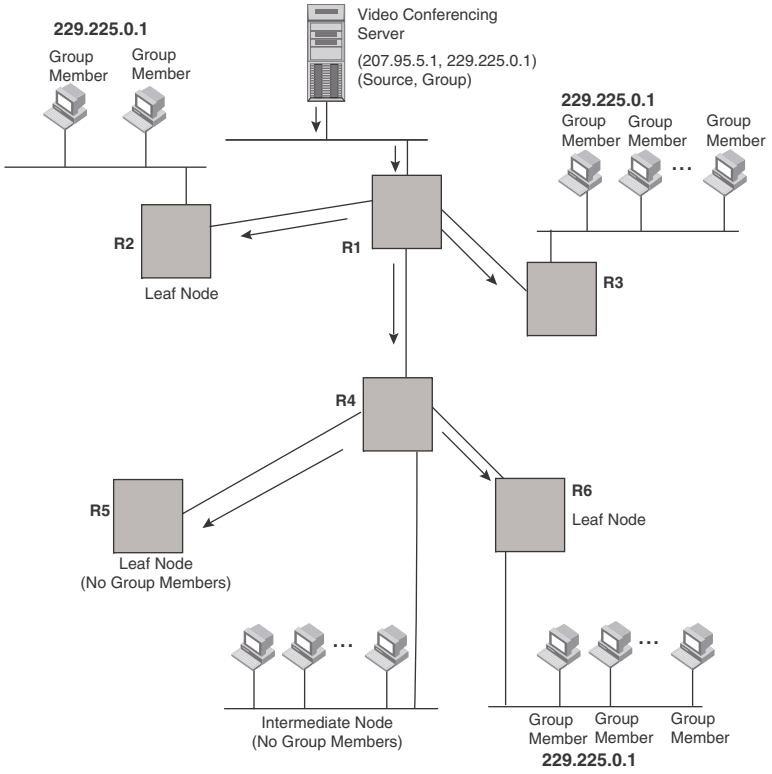
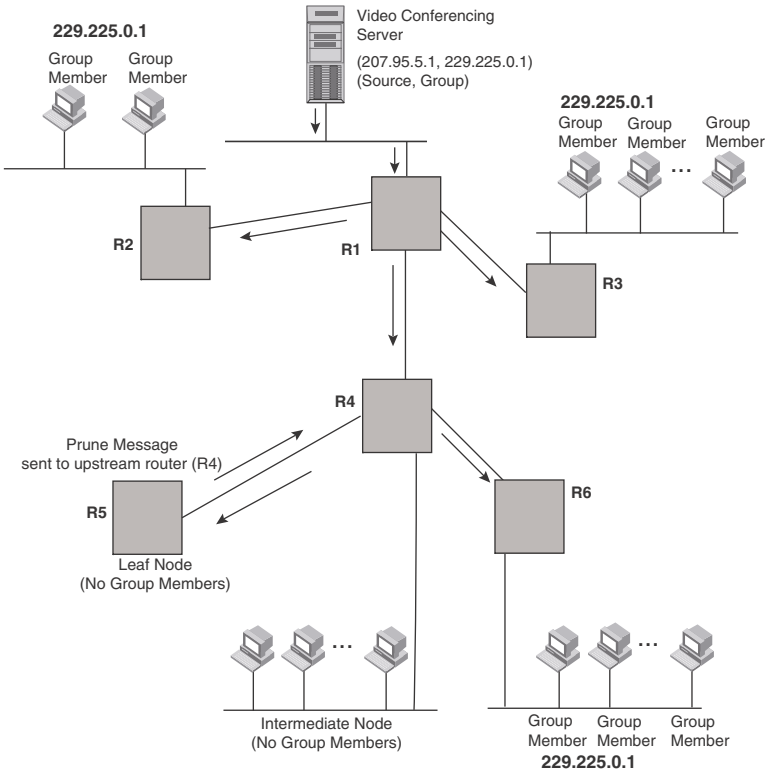


FIGURE 162 Pruning leaf nodes from a multicast tree



Grafts to a multicast tree

A DVMRP router restores pruned branches to a multicast tree by sending graft messages towards the upstream router. Graft messages start at the leaf node and travel up the tree, first sending the message to its neighbor upstream router.

In the example above, if a new 229.255.0.1 group member joins on router R6, which had been pruned previously, a graft will be sent upstream to R4. Since the forwarding state for this entry is in a prune state, R4 sends a graft to R1. Once R4 has joined the tree, it along with R6 will once again receive multicast packets.

You do not need to perform any configuration to maintain the multicast delivery tree. The prune and graft messages automatically maintain the tree.

Configuring DVMRP

Enabling DVMRP globally and on an interface

Suppose you want to initiate the use of desktop video for fellow users on a sprawling campus network. All destination workstations have the appropriate hardware and software but the device's that connect the various buildings need to be configured to support DVMRP multicasts from the designated video conference server as seen in Figure 161.

DVMRP is enabled on each of the PowerConnect devices, shown in Figure 161, on which multicasts are expected. You can enable DVMRP on each PowerConnect independently or remotely from one PowerConnect by a Telnet connection. Follow the same steps for each router.

Globally enabling and disabling DVMRP

To globally enable DVMRP, enter the following command.

```
NetIron(config)# router dvmrp
NetIron(config-dvmrp-router)#
```

Syntax: [no] router dvmrp

- Entering a **router dvmrp** command to enable DVMRP does not require a software reload.
- Entering a **no router dvmrp** command removes all DVMRP configurations on a PowerConnect (**router dvmrp** level) only.

Enabling DVMRP for a specified VRF

To enable DVMRP for the VRF named "blue", use the following commands.

```
NetIron(config)# router dvmrp vrf blue
```

Syntax: [no] router dvmrp [vrf <*vrf-name*>]

The **vrf** parameter allows you to configure DVMRP on the virtual routing instance (VRF) specified by the <*vrf-name*> variable. All DVMRP parameters available for the default router instance are configurable for a VRF-based DVMRP instance:

- Entering a **router dvmrp vrf** command to enable DVMRP does not require a software reload.

- Entering a **no router dvmrp vrf** command removes the DVMRP configuration from the specified VRF only.

Enabling DVMRP on an interface

After globally enabling DVMRP on a PowerConnect, enable it on each interface that will support the protocol.

To enable DVMRP on Router 1 and interface 3, enter the following.

```
NetIron(config)# int e 3/1
NetIron(config-if-e10000-3/1)# ip dvmrp
```

Modifying DVMRP global parameters

DVMRP global parameters come with preset values. The defaults work well in most networks, but you can modify the following global parameters if you need to:

- Neighbor timeout
- Route expire time
- Route discard time
- Prune age
- Graft retransmit time
- Probe interval
- Report interval
- Trigger interval
- Default route

Modifying neighbor timeout

The neighbor timeout specifies the period of time that a router will wait before it defines an attached DVMRP neighbor router as down. Possible values are 60 – 8000 seconds. The default value is 180 seconds.

To modify the neighbor timeout value to 100, enter the following.

```
NetIron(config-dvmrp-router)# nbr 100
```

Syntax: [no] nbr-timeout <*40-8000*>

The default is 180 seconds.

Modifying route expires time

The Route Expire Time defines how long a route is considered valid in the absence of the next route update. Possible values are from 20 – 4000 seconds. The default value is 200 seconds.

To modify the route expire setting to 50, enter the following.

```
NetIron(config-dvmrp-router)# route-expire-timeout 50
```

Syntax: [no] route-expire-timeout <*60-4000*>

Modifying route discard time

The Route Discard Time defines the period of time before a route is deleted. Possible values are from 40 – 8000 seconds. The default value is 340 seconds.

To modify the route discard setting to 150, enter the following.

```
NetIron(config-dvmrp-router)# route-discard-timeout 150
```

Syntax: [no] route-discard-timeout <*40-8000*>

Modifying prune age

The Prune Age defines how long a prune state will remain in effect for a source-routed multicast tree. After the prune age period expires, flooding will resume. Possible values are from 20 – 3600 seconds. The default value is 180 seconds.

To modify the prune age setting to 150, enter the following.

```
NetIron(config-dvmrp-router)# prune 25
```

Syntax: [no] prune-age <*20-3600*>

Modifying graft retransmit time

The Graft Retransmit Time defines the initial period of time that a router sending a graft message will wait for a graft acknowledgement from an upstream router before re-transmitting that message.

Subsequent retransmissions are sent at an interval twice that of the preceding interval. Possible values are from
5 – 3600 seconds. The default value is 10 seconds.

To modify the setting for graft retransmit time to 120, enter the following.

```
NetIron(config-dvmrp-router)# graft 120
```

Syntax: [no] graft-retransmit-time <*5-3600*>

Modifying probe interval

The Probe Interval defines how often neighbor probe messages are sent to the ALL-DVMRP-ROUTERS IP multicast group address. A router's probe message lists those neighbor DVMRP routers from which it has received probes. Possible values are from 5 – 30 seconds. The default value is 10 seconds.

To modify the probe interval setting to 10, enter the following.

```
NetIron(config-dvmrp-router)# probe 10
```

Syntax: [no] probe-interval <*5-30*>

Modifying report interval

The Report Interval defines how often routers propagate their complete routing tables to other neighbor DVMRP routers. Possible values are from 10 – 2000 seconds. The default value is 60 seconds.

To support propagation of DVMRP routing information to the network every 90 seconds, enter the following.

```
NetIron(config-dvmrp-router)# report 90
```

Syntax: [no] report-interval <*10-2000*>

Modifying trigger interval

The Trigger Interval defines how often trigger updates, which reflect changes in the network topology, are sent. Example changes in a network topology include router up or down or changes in the metric. Possible values are from 5 – 30 seconds. The default value is 5 seconds.

To support the sending of trigger updates every 20 seconds, enter the following.

```
NetIron(config-dvmrp-router)# trigger-interval 20
```

Syntax: [no] trigger-interval <*5-30*>

Modifying default route

This defines the default gateway for IP multicast routing.

To define the default gateway for DVMRP, enter the following.

```
NetIron(config-dvmrp-router)# default-gateway 192.35.4.1
```

Syntax: [no] default-gateway <*ip-addr*>

Modifying DVMRP interface parameters

DVMRP global parameters come with preset values. The defaults work well in most networks, but you can modify the following interface parameters if you need to:

- TTL Threshold
- Metric
- Advertising

Modifying the TTL threshold

The TTL defines the minimum value required in a packet in order for the packet to be forwarded out the interface. For example, if the TTL for an interface is set at 10 it means that only those packets with a TTL value of 10 or more are forwarded. Likewise, if an interface is configured with a TTL Threshold value of 1, all packets received on that interface are forwarded. Possible values are from 1 – 64. The default value is 1.

To set a TTL of 64, enter the following.

```
NetIron(config)# int e 1/4
NetIron(config-if-e10000-1/4)# ip dvmrp ttl-threshold 60
```

Syntax: [no] ip dvmrp ttl-threshold <*1-64*>

Modifying the metric

The router uses the metric when establishing reverse paths to some networks on directly attached interfaces. Possible values are from 1 – 31 hops. The default is 1.

To set a metric of 15 for a DVMRP interface, enter the following.

```
NetIron(config)# interface e 3/5
NetIron(config-if-e10000-3/5)# ip dvmrp metric 15
```

Syntax: [no] ip dvmrp metric <*1-31*>

Disabling advertising

You can disable the advertisement of a local route on the interface. By default, advertising is enabled.

To disable advertising on an interface, enter the following.

```
NetIron(config-if-e10000-1/4)# ip dvmrp advertise-local off
```

Syntax: [no] ip dvmrp advertise-local off

Use the **no** option with this command to re-enable advertising.

Displaying DVMRP information

You can display the following DVMRP information:

- DVMRP group information
- DVMRP interface information
- DVMRP multicast cache information
- DVMRP neighbor information
- DVMRP active prune information
- Available multicast resources
- IP multicast route information
- Active multicast traffic information

Displaying DVMRP group information

To display DVMRP group information, enter the following command.

```
NetIron# show ip dvmrp group
Total number of groups: 2
1    Group 225.1.1.1           Ports
     Group member at  e2/3: v30
2    Group 226.1.1.1           Ports
     Group member at  e2/4: v40
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] group

The **vrf** option allows you to display DVMRP group information for the VRF instance identified by the <*vrf-name*> variable.

The display shows the following information.

Table 0.2:

This field...	Displays...
Total Number of Groups	The total number of DVMRP groups in the network.
Group	The DVMRP group address.
Ports	The PowerConnect ports connected to receivers of the groups.

Clearing the DVMRP group membership table

To clear the DVMRP group membership table, enter the following command.

```
NetIron# clear ip dvmrp cache
```

Syntax: clear ip dvmrp [vrf <*vrf-name*>] cache

This command clears the DVMRP membership for the default router instance or for a specified VRF.

Use the **vrf** option to clear the traffic information for a VRF instance specified by the <*vrf-name*> variable.

Displaying DVMRP interface information

To display DVMRP Interface information, enter the following command.

```
NetIron# show ip dvmrp interface
Interface e5/2
TTL Threshold: 1, Enabled, Querier
Local Address: 172.5.1.1
DR: itself
Neighbor:
  172.5.1.2
Interface e8/1
TTL Threshold: 1, Enabled, Querier
Local Address: 172.8.1.1
DR: itself
Interface v10
TTL Threshold: 1, Enabled, Querier
Local Address: 192.1.1.1 logical Vid=1
DR: itself
Interface v20
TTL Threshold: 1, Enabled, Querier
Local Address: 192.2.1.1 logical Vid=2
DR: itself
Interface v30
TTL Threshold: 1, Enabled, Querier
Local Address: 192.3.1.1 logical Vid=3
DR: itself
Interface v40
TTL Threshold: 1, Enabled, Querier
Local Address: 192.4.1.1 logical Vid=4
DR: itself
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] interface

The **vrf** option allows you to display DVMRP interface information for the VRF instance identified by the <*vrf-name*> variable.

The display shows the following information

Table 0.3:

This field...	Displays...
Interface	The type of interface and the interface number. The interface can be one of the following: • Ethernet • VE The number is either a port and slot number or the virtual interface (VE) number.
TTL Threshold	Following the TTL threshold value, the interface state is listed. The interface state can be one of the following: • Disabled • Enabled
Local Address	Indicates the IP address configured on the port or virtual interface
DR	Designated router.

Displaying DVMRP multicast cache information

To display DVMRP Multicast Cache information, enter the following command.

```
NetIron# show ip mcache
Total 2 entries
1    (192.2.1.2, 226.1.1.1) in v20 (e2/2)
     L3 (HW) 1: e2/4(VL40)
     fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 swRepl=0
     age=0 fid: 8012,
2    (192.1.1.2, 225.1.1.1) in v10 (e2/1)
     L3 (HW) 1: e2/3(VL30)
     fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 swRepl=0
     age=0 fid: 8011,
Total number of mcache entries 2
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] mcache

The **vrf** option allows you to display DVMRP Multicast cache information for the VRF instance identified by the <*vrf-name*> variable

The display shows the following information

Table 0.4:

This field...	Displays...
Total number of mcache entries	The total number of entries in the DVMRP multicast cache.

Displaying DVMRP neighbor information

To display DVMRP Neighbor information, enter the following command:

```
NetIron# show ip dvmrp nbr
Port   Phy_p  Neighbor        GenId    Age   UpTime
e5/2   e5/2   172.5.1.2       00000000 170   440
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] nbr

The **vrf** option allows you to display DVMRP neighbor information for the VRF instance identified by the <*vrf-name*> variable.

The display shows the following information.

Table 0.5:

This field...	Displays...
Port	The port and slot number for the neighbors interface.
Phy_p	The physical port and slot number for the neighbors interface.
Neighbor	The IP address of the DVMRP neighbor.
GenId	The neighbor's generation ID.
Age	The minimum time remaining before this DVMRP neighbor is aged out.
UpTime	The time in seconds that the DVMRP neighbor has been up.

Displaying DVMRP active prune information

To display DVMRP Active Prune information, enter the following command.

```
NetIron# show ip dvmrp prune
Port SourceNet       Group           Nbr             Age
                                                     sec
e5/2  192.2.1.2       226.1.1.1      172.5.1.1       60
e5/2  192.1.1.2       225.1.1.1      172.5.1.1       60
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] prune

The **vrf** option allows you to display DVMRP active prune information for the VRF instance identified by the <*vrf-name*> variable.

The display shows the following information.

Table 0.6:

This field...	Displays...
Port	The port and slot number for the DVMRP prune.
SourceNet	The address of the source or source network that has been pruned.
Group	The group address that has been pruned
Nbr	The IP address of the DVMRP neighbor.
Age Sec	The amount of time remaining before this prune expires at the upstream neighbor.

Displaying available multicast resources

To display Available Multicast Resources, enter the following command.

```
NetIron# show ip dvmrp resource
                     allocated    in-use available allo-fail  up-limita
DVMRP route               2048        13      2035         0      2048
route interface           2048        13      2035         0      8192
NBR list                   128         1       127         0      1874
prune list                  64         2        62         0       256
graft list                  64         0        64         0       256
mcache                     128         2       126         0      4096
mcache hash link           547         2       545         0  no-limit
graft if no mcache         197         0       197         0  no-limit
IGMP group                 256         2       254         0      2048
pim/dvm intf. group        256         2       254         0  no-limit
pim/dvm global group       256         2       254         0      4096
HW replic vlan            2000         4      1996         0  no-limit
HW replic port            1024         2      1022         0  no-limit
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] resource

The **vrf** option allows you to display available multicast resources for the VRF instance identified by the <*vrf-name*> variable.

Displaying IP multicast route Information

To display IP multicast route information, enter the following command.

```
NetIron# show ip dvmrp route
P:Parent M:Metric
Total Routes=10
SourceNet        Mask             Gateway         P      M

100.1.1.0       255.255.255.0     172.5.1.2       e5/2   2
 Int e5/2   phy_p e5/2   nbr        172.5.1.2 age=0 NOT downstream
101.1.1.0       255.255.255.0     172.5.1.2       e5/2   2
 Int e5/2   phy_p e5/2   nbr        172.5.1.2 age=0 NOT downstream
102.1.1.0       255.255.255.0     172.5.1.2       e5/2   2
 Int e5/2   phy_p e5/2   nbr        172.5.1.2 age=0 NOT downstream
103.1.1.0       255.255.255.0     172.5.1.2       e5/2   2
 Int e5/2   phy_p e5/2   nbr        172.5.1.2 age=0 NOT downstream
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] route

The **vrf** option allows you to display multicast route information for the VRF instance identified by the <*vrf-name*> variabl.

The display shows the following information.

Table 0.7:

This field...	Displays...
Total Routes	The number of entries in the DVMRP routing table.
SourceNet	The IP address of the source network for the route.
Mask	The subnet mask for the router.
Gateway	The IP address of the gateway for the route.
P	The parent port.
M	The metric which is the distance in hops to the source subnet.

Displaying active multicast traffic information

To display active multicast traffic information, enter the following command.

```
NetIron# show ip dvmrp traffic
Port          Probe                    Graft                    Prune
     [Rx      Tx      Dscrd] [Rx      Tx      Dscrd] [Rx      Tx      Dscrd]
e5/2   111     112     0       0       0       0       9       0       1
e8/1   0       220     0       0       0       0       0       0       0
v10    0       211     0       0       0       0       0       0       0
v20    0       210     0       0       0       0       0       0       0
Total 111     1718     0       0       0       0       9       0       1
IGMP Statistics:
  Total Discard/chksum  0/0
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] traffic

The **vrf** option allows you to display multicast traffic information for the VRF instance identified by the <*vrf-name*> variable.

The display shows the following information.

Table 0.8:

This field...	Displays...
Port	The port or virtual interface on which the PIM interface is configured.
Probe	The number of received, transmitted, and discarded probe messages.
Graft	The number of received, transmitted, and discarded graft messages.
Prune	The number of received, transmitted, and discarded prune messages.
Total Discard or chksum	The total number of IGMP messages discarded, including a separate counter for those that failed the checksum comparison.

Clearing DVMRP traffic statistics

To clear statistics for DVMRP traffic, enter the following command.

```
NetIron# clear ip dvmrp traffic
```

Syntax: clear ip dvmrp [vrf <*vrf-name*>] traffic

This command clears all the dvmrp multicast traffic information on all interfaces on the device or for the interfaces of a specified VRF.

Use the **vrf** option to clear the traffic information for a VRF instance specified by the <*vrf-name*> variable.

Displaying DVMRP Settings

To display global DVMRP settings or DVMRP settings for a specified VRF. To display global IGMP settings, enter the following command.

```
NetIron# show ip dvmrp settings
Global DVMRP Settings
  Version: 3.5
  Probe interval: 10, Neighbor timeout: 180
  Graft Retransmit interval: 10, Prune Lifetime/Age: 180
  Maximum Routes: 0, Active Route Count: 0 (0)
  Route Expire interval: 200, Route Discard interval: 340
  Route Report interval: 60, Trigger Update interval: 5
  Maximum Mcache: 0,  Current Count: 0
```

Syntax: show ip dvmrp [vrf <*vrf-name*>] settings

The **vrf** parameter specifies that you want to display DVMRP settings information for the VRF specified by the <*vrf-name*> variable.

The report shows the following information:

Table 0.9:

This field	Displays
Version	The DVMRP version operating on the router.
Probe Interval	The frequency in seconds at which neighbor probe messages are sent to the ALL-DVMRP-ROUTERS IP multicast group address. Possible values are 5 to 30 seconds. The default value is 10 seconds. This parameter is configured in "Modifying probe interval".
Neighbor timeout	The period of time in seconds that a router will wait before it defines an attached DVMRP neighbor router as down. Possible values are 60 to 8000 seconds and the default value is 180 seconds. This parameter is configured in "Modifying neighbor timeout"
Graft Retransmit Interval	The initial period of time in seconds that a router sending a graft message will wait for a graft acknowledgement from an upstream router before re-transmitting that message. Subsequent retransmissions are sent at an interval twice that of the preceding interval. Possible values are 5 to 3600 seconds and the default value is 120 seconds. This parameter is configured in "Modifying graft retransmit time"
Prune Lifetime or Age	The period of time in seconds that a prune state will remain in effect for a source-routed multicast tree. After the prune age period expires, flooding will resume. Possible values are from 20 to 3600 seconds. The default value is 180 seconds. This parameter is configured in "Modifying prune age"
Maximum Routes	The maximum number of DVMRP routes allowed on the router.
Active Route Count	The number of active DVMRP routes.
Route Expire Interval	The time period in seconds during which a route is considered valid in the absence of the next route update. Possible values are from 20 to 4000 seconds and the default value is 200 seconds.. This parameter is configured in "Modifying route expires time".
Route Discard Interval	The time period in seconds before a route is deleted. Possible values are from 40 to 8000 seconds and the default value is 340 seconds.. This parameter is configured in "Modifying route discard time".

Table 0.9:

This field	Displays
Route Report Interval	The time interval in seconds between which the router propagates its complete routing table to other neighbor DVMRP routers. Possible values are from 10 to 2000 seconds and the default value is 60 seconds.. This parameter is configured in "Modifying report interval".
Trigger Update Interval	The time interval in seconds between which the router sends trigger updates which reflect changes in the network topology. Possible values are from 5 to 30 seconds and the default value is 5 seconds.. This parameter is configured in "Modifying trigger interval"
Maximum Mcache	The maximum number multicast cache entries for DVMRP allowed on the router.
Current Count	The number of multicast cache entries currently used.

Configuring a static multicast route

Static multicast routes allow you to control the network path used by multicast traffic. Static multicast routes are especially useful when the unicast and multicast topologies of a network are different. You can avoid the need to make the topologies similar by instead configuring static multicast routes.

NOTE
This feature is not supported for DVMRP.

You can configure more than one static multicast route. The NetIron always uses the most specific route that matches a multicast source address. Thus, if you want to configure a multicast static route for a specific multicast source and also configure another multicast static route for all other sources, you can configure two static routes as shown in the examples below.

To add static routes to multicast router A (refer to Figure 163), enter commands such as the following.

```
PIMRouterA(config)# ip mroute 207.95.10.0 255.255.255.0 ethernet 1/2 distance 1
PIMRouterA(config)# ip mroute 0.0.0.0 0.0.0.0 ethernet 2/3 distance 1
PIMRouterA(config)# write memory
```

Syntax: [no] ip mroute <*ip-addr*> interface ethernet <*slot*>/<*portnum*> | pos <*slot*>/<*portnum*> |ve <*num*> [distance <*num*>]

Or

Syntax: [no] ip mroute <*ip-addr*> rpf_address <*rpf-num*>

The <*ip-addr*> command specifies the PIM source for the route.

NOTE
In IP multicasting, a route is handled in terms of its source, rather than its destination.

You can use the **ethernet** <*slot*>/<*portnum*> or **pos** <*slot*>/<*portnum*> parameters to specify a physical port or the **ve** <*num*> parameter to specify a virtual interface.

NOTE
The **ethernet** <*slot*>/<*portnum*> and **pos** <*slot*>/<*portnum*>parameters do not apply to PIM SM.

The **distance** <*num*> parameter sets the administrative distance for the route. When comparing multiple paths for a route, the NetIron prefers the path with the lower administrative distance.

NOTE
Regardless of the administrative distances, the NetIron always prefers directly connected routes over other routes.

The **rpf_address** <*rpf-num*> parameter specifies an RPF number.

The example above configures two static multicast routes. The first route is for a specific source network, 207.95.10.0/24. If the NetIron receives multicast traffic for network 207.95.10.0/24, the traffic must arrive on port 1/2. The second route is for all other multicast traffic. Traffic from multicast sources other than 207.95.10.0/24 must arrive on port 2/3.

Figure 163 shows an example of an IP Multicast network. The two static routes configured in the example above apply to this network. The commands in the example above configure PIM router A to accept PIM packets from 207.95.10.0/24 when they use the path that arrives at port 1/2, and accept all other PIM packets only when they use the path that arrives at port 2/3.

The distance parameter sets the administrative distance. This parameter is used by the software to determine the best path for the route. Thus, to ensure that the NetIron uses the default static route, assign a low administrative distance value. When comparing multiple paths for a route, the NetIron prefers the path with the lower administrative distance.

Configuring a static multicast route within a VRF

The Multi-Service IronWare software allows you to configure a static multicast route within a virtual routing instance (VRF). The static multicast route is defined within the VRF configuration as shown in the following.

```
NetIron(config)# ip vrf vpn1
NetIron(config-vrf-vpn1)# ip mroute 105.105.105.0/24 14.14.14.105
```

Syntax: [no] ip vrf <*vrf-name*>

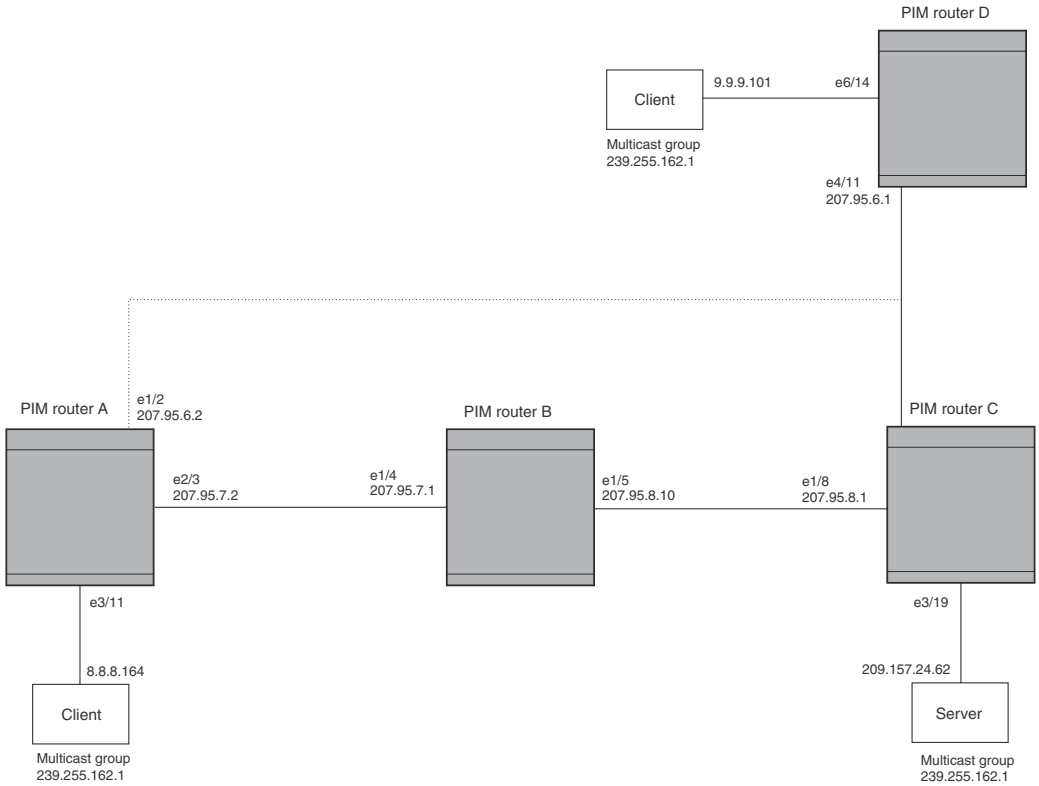
Syntax: [no] ip mroute <*ip-addr*>

The **ip vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

The <*ip-addr*> parameter specifies the destination IP address .

NOTE
Configuring a static multicast route for the default VRF is still accomplished using the command described in "Configuring a static multicast route" on page 1219.

FIGURE 163 Example multicast static routes



To add a static route to a virtual interface, enter commands such as the following.

```
NetIron(config)# mroute 3 0.0.0.0 0.0.0.0 int ve 1 distance 1
NetIron(config)# write memory
```

IGMP V3

The Internet Group Management Protocol (IGMP) allows an IPV4 system to communicate IP Multicast group membership information to its neighboring routers. The routers in turn limit the multicast of IP packets with multicast destination addresses to only those interfaces on the router that are identified as IP Multicast group members.

In IGMP V2, when a router sent a query to the interfaces, the clients on the interfaces respond with a membership report of multicast groups to the router. The router can then send traffic to these groups, regardless of the traffic source. When an interface no longer needs to receive traffic from a group, it sends a leave message to the router which in turn sends a group-specific query to that interface to see if any other clients on the same interface is still active.

In contrast, IGMP V3 provides selective filtering of traffic based on traffic source. A router running IGMP V3 sends queries to every multicast enabled interface at the specified interval. These general queries determine if any interface wants to receive traffic from the router. The following are the three variants of the Query message:

- A "General Query" is sent by a multicast router to learn the complete multicast reception state of the neighboring interfaces. In a General Query, both the Group Address field and the Number of Sources (N) field are zero.

- A "Group-Specific Query" is sent by a multicast router to learn the reception state, with respect to a *single* multicast address, of the neighboring interfaces. In a Group-Specific Query, the Group Address field contains the multicast address of interest, and the Number of Sources (N) field contains zero.
- A "Group-and-Source-Specific Query" is sent by a multicast router to learn if any neighboring interface desires reception of packets sent to a specified multicast address, from any of a specified list of sources. In a Group-and-Source-Specific Query, the Group Address field contains the multicast address of interest, and the Source Address [i] fields contain the source address(es) of interest.

The interfaces respond to these queries by sending a membership report that contains one or more of the following records that are associated with a specific group:

- Current-State Record that indicates from which sources the interface wants to receive and not receive traffic. The record contains source address of interfaces and whether or not traffic will be received or included (IS_IN) or not received or excluded (IS_EX) from that source.
- Filter-mode-change record. If the interface changes its current state from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if an interface's current state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.

  IGMP V2 Leave report is equivalent to a TO_IN (empty) record in IGMP V3. This record means that no traffic from this group will be received regardless of the source.

  An IGMP V2 group report is equivalent to an IS_EX (empty) record in IGMP V3. This record means that all traffic from this group will be received regardless of source.
- Source-List-Change Record. If the interface wants to add or remove traffic sources from its membership report, the membership report can have an ALLOW record, which contains a list of new sources from which the interface wishes to receive traffic. It can also contains a BLOCK record, which lists current traffic sources from which the interfaces wants to stop receiving traffic.

In response to membership reports from the interfaces, the router sends a Group-Specific or a Group-and-Source Specific query to the multicast interfaces. For example, a router receives a membership report with a Source-List-Change record to block old sources from an interface. The router sends Group-and-Source Specific Queries to the source and group (S,G) identified in the record. If none of the interfaces is interested in the (S,G), it is removed from (S,G) list for that interface on the router.

Each IGMP V3-enabled router maintains a record of the state of each group and each physical port within a virtual routing interface. This record contains the group, group-timer, filter mode, and source records information for the group or interface. Source records contain information on the source address of the packet and source timer. If the source timer expires when the state of the group or interface is in Include mode, the record is removed.

Default IGMP version

IGMP V3 is available for PowerConnect Routers; however, these routers are shipped with IGMP V2-enabled. You must enable IGMP V3 globally or per interface.

Also, you can specify what version of IGMP you want to run on a device globally, on each interface (physical port or virtual routing interface), and on each physical port within a virtual routing interface. If you do not specify an IGMP version, IGMP V2 will be used.

Compatibility with IGMP V1 and V2

Different multicast groups, interfaces, and routers can run their own version of IGMP. Their version of IGMP is reflected in the membership reports that the interfaces send to the router. Routers and interfaces must be configured to recognized the version of IGMP you want them to process.

An interface or router sends the queries and reports that include its IGMP version specified on it. It may recognize a query or report that has a different version. For example, an interface running IGMP V2 can recognize IGMP V3 packets, but cannot process them. Also, a router running IGMP V3 can recognize and process IGMP V2 packet, but when that router sends queries to an IGMP V2 interface, the downgraded version is supported, no the upgraded version.

If an interface continuously receives queries from routers that are running versions of IGMP that are different from what is on the interface, the interface logs warning messages in the syslog every five minutes. Reports sent by interfaces to routers that contain different versions of IGMP do not trigger warning messages; however, you can see the versions of the packets using the **show ip igmp traffic** command.

The version of IGMP can be specified globally, per interface (physical port or virtual routing interface), and per physical port within a virtual routing interface. The IGMP version set on a physical port within a virtual routing interface supersedes the version set on a physical or virtual routing interface. Likewise, the version on a physical or virtual routing interface supersedes the version set globally on the device. The sections below present how to set the version of IGMP.

Globally enabling the IGMP version

To globally identify the IGMP version on a Dell device, enter the following command.

```
NetIron(config)# ip igmp version 3
```

Syntax: [no] ip igmp version <*version-number*>

Enter 1, 2, or 3 for <*version-number*>. Version 2 is the default version.

Enabling the IGMP version per interface setting

To specify the IGMP version for a physical port, enter a command such as the following.

```
NetIron(config)# interface eth 1/5
NetIron(config-if-1/5)# ip igmp version 3
```

To specify the IGMP version for a virtual routing interface on a physical port, enter a command such as the following.

```
NetIron(config)# interface ve 3
NetIron(config-vif-1) ip igmp version 3
```

Syntax: [no] ip igmp version <*version-number*>

Enter 1, 2, or 3 for <*version-number*>. Version 2 is the default version.

Enabling the IGMP version on a physical port within a virtual routing interface

To specify the IGMP version recognized by a physical port that is a member of a virtual routing interface, enter a command such as the following.

```
NetIron(config)# interface ve 3
NetIron(config-vif-3)# ip igmp version 2
NetIron(config-vif-3)# ip igmp port-version 3 e1/3 to e1/7 e2/9
```

In this example, the second line sets IGMP V2 on virtual routing interface 3. However, the third line set IGMP V3 on ports 1/3 through 1/7 and port e2/9. All other ports in this virtual routing interface are configured with IGMP V2.

Syntax: [no] ip igmp port-version <*version-number*> ethernet <*port-number*>

Enter 1, 2, or 3 for <*version-number*>. IGMP V2 is the default version.

The **ethernet** <*port-number*> parameter specifies which physical port within a virtual routing interface is being configured.

Enabling membership tracking and fast leave

NOTE
The IGMP V3 fast leave feature is supported in include mode, but does not work in the exclude mode.

IGMP V3 provides membership tracking and fast leave of clients. In IGMP V2, only one client on an interface needs to respond to a router's queries; therefore, some of the clients may be invisible to the router, making it impossible for the switch to track the membership of all clients in a group. Also, when a client leaves the group, the switch sends group specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the switch waits three seconds before it stops the traffic.

IGMP V3 contains the tracking and fast leave feature that you enable on virtual routing interfaces. Once enabled, all physical ports on that virtual routing interface will have the feature enabled. IGMP V3 requires all clients to respond to general and group specific queries so that all clients on an interface can be *tracked. Fast leave* allows clients to leave the group without the three second waiting period, if the following conditions are met:

- If the interface, to which the client belongs, has IGMP V3 clients only. Therefore, all physical ports on a virtual routing interface must have IGMP V3 enabled and no IGMP V1 or V2 clients can be on the interface. (Although IGMP V3 can handle V1 and V2 clients, these two clients cannot be on the interface in order for fast leave to take effect.)
- No other client on the interface is receiving traffic from the group to which the client belongs.

  Every group on the physical interface of a virtual routing interface keeps its own tracking record. It can track by (source, group).

For example, two clients (Client A and Client B) belong to group1 but each is receiving traffic streams from different sources. Client A receives a stream from (source_1, group1) and Client B receives it from (source_2, group1). Now, if Client B leaves, the traffic stream (source_2, group1) will be stopped immediately. The **show ip igmp group tracking** command displays that clients in a group that are being tracked.

If a client sends a leave message, the client is immediately removed from the group. If a client does not send a report during the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

To enable the tracking and fast leave feature, enter commands such as the following.

```
NetIron(config)# interface ve 13
NetIron(config-vif-13)# ip igmp tracking
```

Syntax: [no] ip igmp tracking

Creating a static IGMP group

To configure a physical port to be a permanent (static) member of an IGMP group, enter the following commands.

```
NetIron(config)# interface ethernet 1/5
NetIron(config-if-e1000-1/5)# ip igmp static-group 224.10.1.1
```

Syntax: [no] ip igmp static-group <*ip-address*>

Enter the IP address of the static IGMP group for <*ip-address*>.

To configure a virtual port to be a permanent (static) member of an IGMP group, enter the following commands.

```
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ip igmp static-group 224.10.1.1 ethernet 1/5
```

Syntax: [no] ip igmp static-group <*ip-address*> ethernet <*slot-number*>/<*port-number*>

Enter the IP address of the static IGMP group for <*ip-address*>.

Enter the ID of the physical port of the VLAN that will be a member of the group for **ethernet** <*slot-number*>/<*port-number*>.

NOTE
IGMPv3 does not support static IGMP group members.

NOTE
Static IGMP groups are supported only in Layer 3 mode.

Setting the query interval

The IGMP query interval period defines how often a switch will query an interface for group membership. Possible values are 2-3600 seconds and the default value is 125 seconds, but the value you enter must be a little more than twice the group membership time.

To modify the default value for the IGMP query interval, enter the following.

```
NetIron(config)# ip igmp query-interval 120
```

Syntax: [no] ip igmp query-interval <*2-3600*>

The interval must be a little more than two times the group membership time.

Setting the group membership time

Group membership time defines how long a group will remain active on an interface in the absence of a group report. Possible values are from 5 – 26000 seconds and the default value is 260 seconds.

To define an IGMP membership time of 240 seconds, enter the following.

```
NetIron(config)# ip igmp group-membership-time 240
```

Syntax: [no] ip igmp group-membership-time <*5-26000*>

Setting the maximum response time

The maximum response time defines the maximum number of seconds that a client can wait before it replies to the query sent by the router. Possible values are 1 – 25. The default is 10.

To change the IGMP maximum response time, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# ip igmp max-response-time 8
```

Syntax: [no] ip igmp max-response-time <*num*>

The <*num*> parameter specifies the maximum number of seconds for the response time. Enter a value from 1 – 25. The default is 10.

Displaying IGMPv3 information

The sections below present the show commands available for IGMP V3.

Displaying IGMP group status

You can display the status of all IGMP multicast groups on a device by entering the following command.

```
NetIron# show ip igmp group
Total 2 entries
------------------------------------------------------
Idx Group Address    Port   Intf   Mode    Timer Srcs
---+----------------+------+------+-------+-----+----
  1 232.0.0.1        e6/2   v30    include     0   7
  2 226.0.0.1        e6/2   v30    exclude   240   2
                     e6/3   e6/3   include     0   3
Total number of groups 2
```

To display the status of one IGMP multicast group, enter a command such as the following.

```
NetIron# show ip igmp group 239.0.0.1 detail
Total 2 entries
------------------------------------------------------
Idx Group Address    Port   Intf   Mode    Timer Srcs
---+----------------+------+------+-------+-----+----
  1 226.0.0.1        e6/2   v30    exclude   218   2
     S: 40.40.40.12
     S: 40.40.40.11
     S: 40.40.40.10
     S: 40.40.40.2          (Age: 218)
     S: 40.40.40.3          (Age: 218)
    226.0.0.1        e6/3   e6/3   include     0   3
     S: 30.30.30.3          (Age: 165)
     S: 30.30.30.2          (Age: 165)
     S: 30.30.30.1          (Age: 165)
```

If the tracking and fast leave feature is enabled, you can display the list of clients that belong to a particular group by entering commands such as the following.

```
NetIron# show ip igmp group 224.1.10.1 tracking
Total 2 entries
----------------------------------------------------
Idx Group Address    Port   Intf   Mode    Timer Srcs
---+---------------+------+------+-------+-----+----
  1 226.0.0.1        e6/2   v30    exclude   253   3
     S: 40.40.40.12
     S: 40.40.40.11
     S: 40.40.40.10
     S: 40.40.40.2          (Age: 253)
                    C: 10.10.10.1       (Age: 253)
    S: 40.40.40.3          (Age: 253)
                    C: 10.10.10.1       (Age: 253)
    226.0.0.1       e6/3   e6/3   include     0   3
     S: 30.30.30.3          (Age: 196)
                    C: 10.2.0.1         (Age: 196)
     S: 30.30.30.2          (Age: 196)
                    C: 10.2.0.1         (Age: 196)
     S: 30.30.30.1          (Age: 196)
                    C: 10.2.0.1         (Age: 196)
```

Syntax: show ip igmp [vrf <*vrf-name*>] group [<*group-address*> [detail] [tracking]]

If you want a report for a specific multicast group, enter that group's address for <*group-address*>. Omit the <*group-address*> if you want a report for all multicast groups.

The **vrf** parameter specifies that you want to display IGMP group information for the VRF specified by the <*vrf-name*> variable.

Enter **detail** if you want to display the source list of the multicast group.

Enter **tracking** if you want information on interfaces that have tracking enabled.

IGMP V2 and V3 statistics displayed on the report for each interface.

Table 0.10:

This field	Displays
Group	The address of the multicast group
Port	The physical port on which the multicast group was received.
Intf	The virtual interface on which the multicast group was received.
Timer	Shows the number of seconds the interface can remain in exclude mode. An exclude mode changes to include mode if it does not receive an "IS_EX" or "TO_EX" message during a certain period of time. The default is 140 seconds.
Mode	Indicates current mode of the interface: include or exclude. If the interface is in Include mode, it admits traffic only from the source list. If an interface is in exclude mode, it denies traffic from the source list and accepts the rest.
Srcs	Identifies the source list that will be included or excluded on the interface. If IGMP V2 group is in exclude mode with a #_src of 0, the group excludes traffic from 0 (zero) source list, which means that all traffic sources are included.

Clearing the IGMP group membership table

To clear the IGMP group membership table, enter the following command.

```
NetIron# clear ip igmp cache
```

Syntax: clear ip igmp [vrf <*vrf-name*>] cache

This command clears the IGMP membership for the default router instance or for a specified VRF.

Use the **vrf** option to clear the traffic information for a VRF instance specified by the <*vrf-name*> variable.

Displaying static IGMP groups

The following command displays static IGMP groups for the “eng” VRF.

```
NetIron#show ip igmp vrf eng static
Group Address    Interface Port List
---------------+---------+---------
      229.1.0.12       4/1 ethe 4/1
      229.1.0.13       4/1 ethe 4/1
      229.1.0.14       4/1 ethe 4/1
      229.1.0.92       4/1 ethe 4/1
```

Syntax: show ip igmp [vrf <*vrf-name*>] static

The **vrf** parameter specifies that you want to display static IGMP group information for the VRF specified by the <*vrf-name*> variable.

Table 0.11:

This field	Displays
Group Address	The address of the multicast group.
Interface Port List	The physical ports on which the multicast groups are received.

Displaying the IGMP status of an interface

You can display the status of a multicast enabled port by entering a command such as the following.

```
NetIron# show ip igmp interface
---------+------+---------+---------------+---------+-----+-----+---------
Intf/Port|Groups| Version |Querier        |  Timer  |V1Rtr|V2Rtr|Tracking
         |      |Oper  Cfg|               |OQrr GenQ|     |     |
---------+------+----+----+---------------+----+----+-----+-----+---------
e6/3           1    3    3 Self              0   94 No    No     Disabled
e6/4           0    2    - Self              0   94 No    No     Disabled
v30            1    3    3                                       Disabled
     e6/2           3    - Self              0   20 No    No
v40            0    3    3                                       Disabled
     e6/2           3    - Self              0   20 No    No
v50            0    2    -                                       Disabled
    e12/1           2    - Self              0   29 No    No
     e6/8           2    - 50.1.1.10        46    0 No    Yes
     e6/1           2    - Self              0  115 No    Yes
```

Syntax: show ip igmp [vrf <*vrf-name*>] interface [ve <*number*> | ethernet <*slot/port*> | pos <*slot/port*> | tunnel <*num*>]

The **vrf** parameter specifies that you want to display IGMP interface information for the VRF specified by the <*vrf-name*> variable.

Enter **ve** and its <*number*>, **pos** and its <*slot/port*> or **ethernet** and its <*slot/port*> to display information for a specific virtual routing interface, pos interface or ethernet interface.

The **tunnel** <*num*> parameter specifies a GRE tunnel interface that is being configured. The GRE tunnel interface is enabled under the router PIM configuration.

Entering an address for <*group-address*> displays information for a specified group on the specified interface.

The report shows the following information:

Table 0.12:

This field	Displays
Intf	The virtual interface on which IGMP is enabled.
Port	The physical port on which IGMP is enabled.
Groups	The number of groups that this interface or port has membership.
Version	
Oper	The IGMP version that is operating on the interface.
Cfg	The IGMP version that is configured for this interface.
Querier	Where the Querier resides: The IP address of the router where the querier is located or Self – if the querier is on the same router as the intf or port.
Max response	
oQrr	Other Querier present timer.
GenQ	General Query timer
V1Rtr	Whether IGMPv1 is present on the intf or port.
V2Rtr	Whether IGMPv2 is present on the intf or port.
Tracking	Fast tracking status: Enabled or Disabled

Displaying IGMP traffic status

To display the traffic status on each virtual routing interface, enter the following command.

```
NetIron# show ip igmp traffic
Recv  QryV2 QryV3 G-Qry GSQry MbrV2 MbrV3 Leave  IsIN  IsEX ToIN ToEX ALLOW BLK
v5       29     0     0     0     0     0     0     0     0    0    0     0    0
v18      15     0     0     0     0    30     0    60     0    0    0     0    0
v110      0     0     0     0     0    97     0   142    37    2    2     3    2
Send  QryV1 QryV2 QryV3 G-Qry GSQry
v5       0     2     0     0     0
v18      0     0    30    30     0
v110     0     0    30    44    11
```

Syntax: show ip igmp [vrf <*vrf-name*>] traffic

The **vrf** parameter specifies that you want to display IGMP traffic information for the VRF specified by the <*vrf-name*> variable.

The report shows the following information:

Table 0.13:

This field	Displays
QryV2	Number of general IGMP V2 query received or sent by the virtual routing interface.
QryV3	Number of general IGMP V3 query received or sent by the virtual routing interface.
G-Qry	Number of group specific query received or sent by the virtual routing interface.
GSQry	Number of source specific query received or sent by the virtual routing interface.
MbrV2	The IGMP V2 membership report.
MbrV3	The IGMP V3 membership report.
Leave	Number of IGMP V2 “leave” messages on the interface. (See ToEx for IGMP V3.)
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from exclude to include.
ToEX	Number of times the interface mode changed from include to exclude.
ALLOW	Number of times that additional source addresses were allowed or denied on the interface:
BLK	Number of times that sources were removed from an interface.

Clearing IGMP traffic statistics

To clear statistics for IGMP traffic, enter the following command.

```
NetIron# clear ip igmp traffic
```

Syntax: clear ip igmp [vrf <*vrf-name*>] traffic

This command clears all the multicast traffic information on all interfaces on the device.

Use the **vrf** option to clear the traffic information for a VRF instance specified by the <*vrf-name*> variable. T

Displaying IGMP settings

To display global IGMP settings or IGMP settings for a specified VRF. To display global IGMP settings, enter the following command.

```
NetIron# show ip igmp  settings
IGMP Global Configuration
  Query Interval            : 125s
  Configured Query Interval : 125s
  Max Response Time         : 10s
  Group Membership Time     : 260s
  Configured Version        : 2
  Operating Version         : 2
```

Syntax: show ip igmp [vrf <*vrf-name*>] settings

The **vrf** parameter specifies that you want to display IGMP settings information for the VRF specified by the <*vrf-name*> variable.

The report shows the following information:

Table 0.14:

This field	Displays
Query Interval	How often the router will query an interface for group membership.
Configured Query Interval	The query interval that has been configured for the router.
Max Response Time	The length of time in seconds that the router will wait for an IGMP (V1 or V2) response from an interface before concluding that the group member on that interface is down and removing it from the group.
Group Membership Time	The length of time in seconds that a group will remain active on an interface in the absence of a group report.
Configured Version	The IGMP version configured on the router.
Operating Version	The IGMP version operating on the router.

Source-specific multicast

Using the Any-Source Multicast (ASM) service model, sources and receivers register with a multicast address. The protocol uses regular messages to maintain a correctly configured broadcast network where all sources can send data to all receivers and all receivers get broadcasts from all sources.

With Source-specific multicast (SSM), the “channel” concept is introduced where a “channel” consists of a single source and multiple receivers who specifically register to get broadcasts from that source. Consequently, receivers are not burdened with receiving data they have no interest in, and network bandwidth requirements are reduced because the broadcast need only go to a sub-set of users. The address range 232/8 has been assigned by the Internet Assigned Numbers Authority (IANA) for use with SSM.

IGMP V3 and source specific multicast protocols

When IGMP V3 and PIM Sparse (PIM-SM) is enabled, the source specific multicast service (SSM) can be configured. SSM simplifies PIM-SM by eliminating the RP and all protocols related to the RP. IGMPv3 and PIM-SM must be enabled on any ports that you want SSM to operate.

Configuring PIM SSM group range

PIM Source Specific Multicast (SSM) is a subset of the PIM SM protocol. In PIM SSM mode, the shortest path tree (STP) is created at the source. The STP is created between the receiver and source, but the STP is built without the help of the RP. The router closest to the interested receiver host is notified of the unicast IP address of the source for the multicast traffic. PIM SSM goes directly to the source-based distribution tree without the need of the RP connection. PIM SSM is different from PIM SM because it forms its own STP tree, without forming a shared tree. The multicast address group range is 232.0.0.0/8.

To configure a single SSM group address, enter the following command under the router pim configuration:

```
NetIron(config)#router pim
NetIron(config-pim-router)#ssm-enable range 232.1.1.1/8
```

Syntax: [no] ssm-enable range <*group-address*><*address-mask*>

The <*group-address*> parameter specifies the multicast address for the SSM address range .If this is not configured, the range will default to 232/8 as assigned by the Internet Assigned Numbers Authority (IANA) for use with SSM.

The< *address-mask*> parameter specifies the mask for the SSM address range.

To disable SSM, use the [no] form of this command.

Displaying source-specific multicast configuration information

To display PIM Sparse configuration information, use the **show ip pim sparse** command as described in"Displaying basic PIM Sparse configuration information" on page 1174.

Configuring multiple SSM group ranges

The **ssm-enable range** <*acl-id/acl-name*> command allows you to configure multiple SSM group ranges using an ACL.

Configuration Considerations

- The existing **ssm-enable range** <*group-address*><*address-mask*> command will continue to exist.
- The ACL must be configured with the SSM group address in the permit clause of the **ssm-enable range** <*acl-id or acl-name*> command. If the **ssm-enable range** <*group-address*><*address-mask*> command permits a clause, then that group will also operate in the PIM-SM mode.
- If the **ssm-enable range** <*acl-id or acl-name*> command is configured with a non-existent or empty ACL, then the SSM group will operate in PIM-SM mode (non PIM-SSM mode). However when an ACL is added or updated, then the group will exist in a PIM-SSM mode. By default, an empty ACL will deny all.
- By default, the group address mentioned in the IGMPv2 ssm-mapping ACL will decide if the group address is a PIM-SSM group or non PIM-SSM group. Therefore, if a user wants to prevent a group from operating in PIM-SSM mode, then the user's configuration must consistently deny the group in all configuration options for PIM-SSM range.

- ACL of any type (named or unnamed, standard or extended) can be used to specify the SSM group range. If an extended ACL is used, then the destination ip address should be used to specify the group address. Any configuration in the source address of an extended ACL is ignored. Only permit statements are considered in the ACL configuration. Any deny statements in the ACL clause are also ignored.

To configure multiple SSM group address using an ACL, enter the following command under the router pim configuration:

```
NetIron(config)#router pim
NetIron(config-pim-router)#ssm-enable range xyz
```

The example displayed above configures PIM so that it uses the group addresses allowed by ACL, xyz as its PIM SSM range.

Syntax: [no] ssm-enable range <*acl-id or acl-name*>

The <*acl-id/acl-name*> parameter specifies the ACL id or name used to configure multiple SSM group ranges.

To disable the SSM mapping range ACL, use the [no] form of this command.

NOTE
The **ssm-enable range** <*acl-id or acl-name*> command also supports IPv6 traffic. The **ssm-enable range** <*acl-id or acl-name*> command must be configured under the IPv6 router pim configuration to support IPv6.

Displaying information for PIM SSM range ACL

To display information for PIM SSM range ACL configuration enter the following command at any CLI level:

```
NetIron#show ip pim sparse

Global PIM Sparse Mode Settings
 Maximum Mcache         : 0          Current Count              : 0
  Hello interval        : 30          Neighbor timeout          : 105
  Join/Prune interval   : 60          Inactivity interval       : 180
  Register Suppress Time : 60         Register Probe Time       : 10
  SPT Threshold         : 1           Hardware Drop Enabled     : Yes
  Bootstrap Msg interval : 60         Candidate-RP Msg interval  : 60
  Register Stop Delay   : 60          Register Suppress interval : 60
  SSM Enabled           : Yes
  SSM Group Range       : 224.1.1.1/24
  SSM Group Range ACL   : xyz
  Route Precedence      : mc-non-default mc-default uc-non-default uc-default
```

NOTE
The **show ipv6 pim sparse** command also displays PIM SSM range ACL configuration.

IGMPv2 SSM mapping

The PIM-SSM feature requires all IGMP hosts to send IGMPv3 reports. Where you have an IGMPv2 host, this can create a compatibility problem. In particular, the reports from an IGMPv2 host contain a Group Multicast Address but do not contain source addresses. The IGMPv3 reports contain both the Group Multicast Address and one or more source addresses. This feature converts IGMPv2 reports into IGMPv3 reports through use of the **ip igmp ssm-map** commands and a properly configured ACL.

The ACL used with this feature filters for the Group Multicast Address. The ACL is then associated with one or more source addresses using the **ip igmp ssm-map static** command. When the **ip igmp ssm-map enable** command is configured, IGMPv3 reports are sent for IGMPv2 hosts.

The following sections describe how to configure the ACL and the **ip igmp ssm-map** commands to use the IGMPv2 SSM mapping feature:

- Configuring an ACL for IGMPv2 SSM mapping
- Configuring the IGMPv2 SSM Mapping Commands

NOTE
IGMPv2 SSM Mapping is not supported for IGMP static groups.

Configuring an ACL for IGMPv2 SSM mapping

You can use either a standard or extended ACL to identify the group multicast address you want to add source addresses to when creating a IGMPv3 report.

For standard ACLs, you must create an ACL with a permit clause and the **ip-source-address** variable must contain the group multicast address. This can be configured directly with a subnet mask or with the **host** keyword in which case a subnet mask of all zeros (0.0.0.0) is implied.

In the following example, **access-list 20** is configured for the group multicast address: 224.1.1.0 with a subnet mask of 0.0.0.255.

```
NetIron(config)# access-list 20 permit 224.1.1.0 0.0.0.225
```

In the following example, **access-list 20** is configured for the group multicast address: 239.1.1.1 by including the **host** keyword.

```
NetIron(config)# access-list 20 host 239.1.1.1
```

For extended ACLs, the **source address** variable must contain either **000** or the **any** keyword. Additionally, the extended ACL must be configured with a **permit** clause and the host keyword. This can be configured directly with a subnet mask or with the **host** keyword in which case a subnet mask of all zeros (0.0.0.0) is implied.

The **ip-destination-address** variable must contain the group multicast address.

In the following example, **access-list 100** is configured for the group multicast address: 232.1.1.1 with a subnet mask of 0.0.0.255.

```
NetIron(config)# access-list 20 permit 224.1.1.0 0.0.0.225
```

In the following example, **access-list 100** is configured for the group multicast address: 232.1.1.1.

```
NetIron(config)# access-list 100 permit any host 232.1.1.1
```

Configuring the IGMPv2 SSM mapping commands

The **ip ssm-map** commands are used to enable the IGMPv2 mapping feature and to define the maps between IGMPv2 Group addresses and multicast source addresses as described in the following sections.

Enabling IGMPv2 SSM mapping

To enable the IGMPv2 mapping feature enter the command as shown in the following.

```
NetIron(config)# ip igmp ssm-map enable
```

Syntax: [no] ip igmp ssm-map enable

The **no** option is used to turn off the IGMPv2 mapping feature that has previously been enabled.

Configuring the map between a IGMPv2 group address and a multicast source

To configure a map between an IGMPv2 Group address and a multicast source address use the **ip igmp ssm-map static** command, as shown in the following.

```
NetIron(config)# ip igmp ssm-map static 20 1.1.1.1
```

Syntax: [no] ip igmp ssm-map static <*acl-number*> <*source-address*>

The <*acl-number*> variable specifies the ACL that contains the group multicast address.

The <*source-address*> variable specifies the source address that you want to map to the group multicast address specified in the ACL.

The **no** option is used to delete a previously configured SSM map.

Example configuration

In the following example configuration, one extended ACL and two standard ACLs are defined with group multicast addresses. The **ip igmp ssm-map** commands are configured to map the ACLs to source addresses and to enable the feature on the router.

```
NetIron(config)# access-list 20 host 239.1.1.1
NetIron(config)# access-list 20 permit 224.1.1.0 0.0.0.225
NetIron(config)# access-list 100 permit any host 232.1.1.1
NetIron(config)# ip igmp ssm-map static 20 1.1.1.1
NetIron(config)# ip igmp ssm-map static 20 2.2.2.2
NetIron(config)# ip igmp ssm-map static 100 1.1.1.1
NetIron(config)# ip igmp ssm-map enable
```

Displaying an IGMP SSM mapping information

The **show ip igmp ssm-map** command displays the association between a configured ACL and source address mapped to it, as shown in the following.

```
NetIron# show ip igmp ssm-map
+---------+-----------------+
| Acl id  | Source Address  |
+---------+-----------------+
        20          1.1.1.1
       100          1.1.1.1
        20          2.2.2.2
        20          2.2.2.3
```

```
        20          2.2.2.4
        20          2.2.2.5
        20          2.2.2.6
```

Syntax: show ip igmp ssm-map

The **show ip igmp ssm-map** < *group-address*> displays the ACL ID that has the specified multicast group address in its permit list and lists the source addresses mapped to the specified multicast group address , as shown in the following.

```
NetIron# show ip igmp ssm-map 232.1.1.1
+---------+-----------------+
| Acl id  | Source Address  |
+---------+-----------------+
        20          1.1.1.1
       100          1.1.1.1
        20          2.2.2.2
        20          2.2.2.3
        20          2.2.2.4
        20          2.2.2.5
        20          2.2.2.6
```

Syntax: show ip igmp ssm-map <*group-address*>

IP multicast traffic reduction

In Layer 2 mode, by default, the PowerConnect forwards all IP multicast traffic out all ports except the port on which the traffic was received. Forwarding decisions are based on the Layer 2 information in the packets. To reduce multicast traffic through the device, you can enable IP Multicast Traffic Reduction. When this feature is enabled, forwarding decisions are made in hardware, based on multicast group. The device will forward multicast traffic only on the ports attached to multicast group members, instead of forwarding all multicast traffic to all ports.

By default, the device broadcasts traffic addressed to an IP multicast group that does not have any entries in the IGMP table. When you enable IP Multicast Traffic Reduction, the device determines the ports that are attached to multicast group members based on entries in the IGMP table. The IGMP table entries are created when the VLAN receives a group membership report for a group. Each entry in the table consists of an IP multicast group address and the ports from which the device has received Group Membership reports.

When the device receives traffic for an IP multicast group, the device looks in the IGMP table for an entry corresponding to that group. If the device finds an entry, it forwards the group traffic out the ports listed in the corresponding entries, as long as the ports are members of the same VLAN. If the table does not contain an entry corresponding to the group, or if the port is a member of the default VLAN, the device broadcasts the traffic.

Configuration requirements

Consider the following configuration requirements and application notes:

- The IP Multicast Traffic Reduction feature is applicable to Layer 2 mode only.
- If the **route-only** feature is enabled on the PowerConnect, then IP Multicast Traffic Reduction will not be supported.

- This feature is not supported on the default VLAN of the PowerConnect.
- When one or more PowerConnect devices are running Layer 2 IP Multicast Traffic reduction, configure one of the devices for active IGMP and leave the other devices configured for passive IGMP. However, if the IP multicast domain contains a multicast-capable router, configure all the PowerConnect devices for passive IGMP and allow the router to actively send the IGMP queries.
- IP multicast traffic reduction and PIM SM Traffic Snooping are supported on the PowerConnect.

Configuring IP multicast traffic reduction

When you enable IP Multicast Traffic Reduction, you also can configure the following features:

- **IGMP mode** – When you enable IP Multicast Traffic Reduction, the device passively listens for IGMP Group Membership reports by default. If the multicast domain does not have a router to send IGMP queries to elicit these Group Membership reports, you can enable the device to actively send the IGMP queries. The IGMP passive mode is also known as IGMP snooping and facilitates IP Multicast Traffic Reduction.
- **Query interval** – The query interval specifies how often the device sends Group Membership queries. This query interval applies only to the active IGMP mode. The default is 60 seconds. You can change the interval to a value from 10 – 600 seconds.
- **Age interval** – The age interval specifies how long an IGMP group can remain in the IGMP group table without the device receiving a Group Membership report for the group. If the age interval expires before the device receives another Group Membership report for the group, the device removes the entry from the table. The default is 140 seconds. You can change the interval to a value from 10 – 1220 seconds.

Furthermore, when you enable IP Multicast Traffic Reduction, the device forwards all IP multicast traffic by default, but you can enable the device to do the following:

- Forward IP multicast traffic only for groups for which the device has received a Group Membership report.
- Drop traffic for all other groups.

The following sections describe how to configure IP multicast traffic reduction and PIM SM Traffic Snooping parameters on a PowerConnect.

Enabling IP multicast traffic reduction

To enable IP Multicast Traffic Reduction, enter the following command.

```
NetIron(config)# ip multicast
```

Syntax: [no] ip multicast active | passive

When you enable IP multicast on a PowerConnect, all ports on the device are configured for IGMP.

The **active** mode enables all ports to send IGMP queries and receive IGMP reports. I

The **passive** mode enables all ports to receive IGMP queries.

IP Multicast Traffic Reduction cannot be disabled on individual ports of a PowerConnect. IP Multicast Traffic Reduction must be disabled globally by entering the **no ip multicast** command.

To verify that IP Multicast Traffic Reduction is enabled, enter the following command at any level of the CLI.

```
NetIron(config)# show ip multicast
IP multicast is enabled - Active
```

Syntax: show ip multicast

Changing the IGMP mode

When you enable IP Multicast Traffic Reduction on the device, IGMP also is enabled. The device uses IGMP to maintain a table of the Group Membership reports received by the device. You can use active or passive IGMP mode. There is no default mode.

The active and passive IGMP modes are described as follows:

- **Active** – When active IGMP mode is enabled, a Dell device actively sends out IGMP queries to identify IP multicast groups on the network and makes entries in the IGMP table based on the Group Membership reports received from the network.

  NOTE
  Routers in the network generally handle this operation. Use the active IGMP mode only when the device is in a stand-alone Layer 2 Switched network with no external IP multicast router attachments. In this case, enable the active IGMP mode on only one of the devices and leave the other devices configured for passive IGMP mode.

- **Passive** – When passive IGMP mode is enabled, the device listens for IGMP Group Membership reports but does not send IGMP queries. The passive mode is sometimes called "IGMP snooping". Use this mode when another device in the network is actively sending queries.

To enable active IGMP, enter the following command.

```
NetIron(config)# ip multicast active
```

Syntax: [no] ip multicast active | passive

To enable passive IGMP, enter the following command.

```
NetIron(config)# ip multicast passive
```

Modifying the query interval

If IP Multicast Traffic Reduction is set to active mode, you can modify the query interval, which specifies how often a PowerConnect enabled for active IP Multicast Traffic Reduction sends group membership queries.

NOTE
The query interval applies only to the active mode of IP Multicast Traffic reduction.

To modify the query interval, enter a command such as the following.

```
NetIron(config)# ip multicast query-interval 120
```

Syntax: [no] ip multicast query-interval <*interval*>

The <*interval*> parameter specifies the interval between queries. You can specify a value from 10 – 600 seconds. The default is 125 seconds.

Modifying the age interval

When the device receives a Group Membership report, the device makes an entry in the IGMP group table for the group in the report. The age interval specifies how long the entry can remain in the table without the device receiving another Group Membership report.

To modify the age interval, enter a command such as the following.

```
NetIron(config)# ip multicast age-interval 280
```

Syntax: [no] ip multicast age-interval <*interval*>

The <*interval*> parameter specifies the interval between queries. You can specify a value from 10 – 1220 seconds. The default is 260 seconds.

Filtering multicast groups

By default, the PowerConnect forwards multicast traffic for all valid multicast groups. You can configure a PowerConnect to filter out all multicast traffic for groups other than the ones for which the device has received Group Membership reports.

When the device starts up, it forwards all multicast groups even though multicast traffic filters are configured. This process continues until the device receives a group membership report. Once the group membership report is received, the device drops all multicast packets for groups other than the ones for which the device has received the group membership report.

To enable IP multicast filtering, enter the following command.

```
NetIron(config)# ip multicast filter
```

Syntax: [no] ip multicast filter

NOTE
When IGMP snooping is enabled on the multicast instance, the traffic will be forwarded to router ports although multicast filter is configured. Also, the ip multicast filter command is used only to avoid flooding. In case of PIM snooping, traffic is dropped since there are no router ports.

PIM SM traffic snooping

By default, when a PowerConnect receives an IP multicast packet, the device does not examine the multicast information in the packet. Instead, the device simply forwards the packet out all ports except the port that received the packet. In some networks, this method can cause unnecessary traffic overhead in the network. For example, if the PowerConnect is attached to only one group source and two group receivers, but has devices attached to every port, the device forwards group traffic out all ports in the same broadcast domain except the port attached to the source, even though there are only two receivers for the group.

PIM SM traffic snooping eliminates the superfluous traffic by configuring the device to forward IP multicast group traffic only on the ports that are attached to receivers for the group.

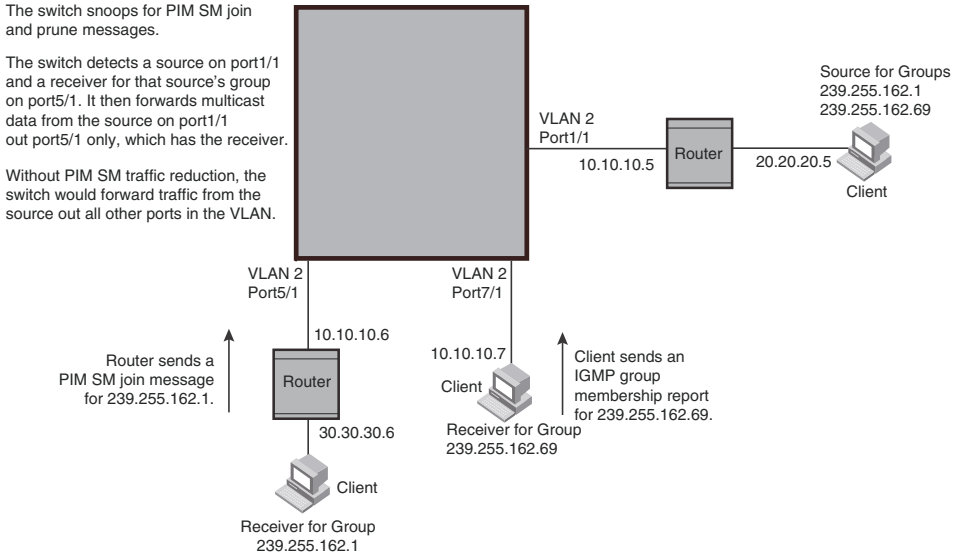
PIM SM traffic snooping requires IP multicast traffic reduction to be enabled on the device. IP multicast traffic reduction configures the device to listen for IGMP messages. PIM SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for PIM SM join and prune messages sent from one PIM SM router to another through the device.

NOTE
This feature applies only to PIM SM version 2 (PIM V2).

Application examples

Figure 164 shows an example application of the PIM SM traffic snooping feature. In this example, a device is connected through an IP router to a PIM SM group source that is sending traffic for two PIM SM groups. The device also is connected to a receiver for each of the groups.

FIGURE 164 PIM SM traffic reduction in enterprise network



When PIM SM traffic snooping is enabled, the device starts listening for PIM SM join and prune messages and IGMP group membership reports. Until the device receives a PIM SM join message or an IGMP group membership report, the device forwards IP multicast traffic out all ports. Once the device receives a join message or group membership report for a group, the device forwards subsequent traffic for that group only on the ports from which the join messages or IGMP reports were received.

In this example, the router connected to the receiver for group 239.255.162.1 sends a join message toward the group's source. Since PIM SM traffic snooping is enabled on the device, the device examines the join message to learn the group ID, then makes a forwarding entry for the group ID and the port connected to the receiver's router. The next time the device receives traffic for 239.255.162.1 from the group's source, the device forwards the traffic only on port 5/1, since that is the only port connected to a receiver for the group.

Notice that the receiver for group 239.255.162.69 is directly connected to the device. As result, the device does not see a join message on behalf of the client. However, since IP multicast traffic reduction also is enabled, the device uses the IGMP group membership report from the client to select the port for forwarding traffic to group 239.255.162.69 receivers.

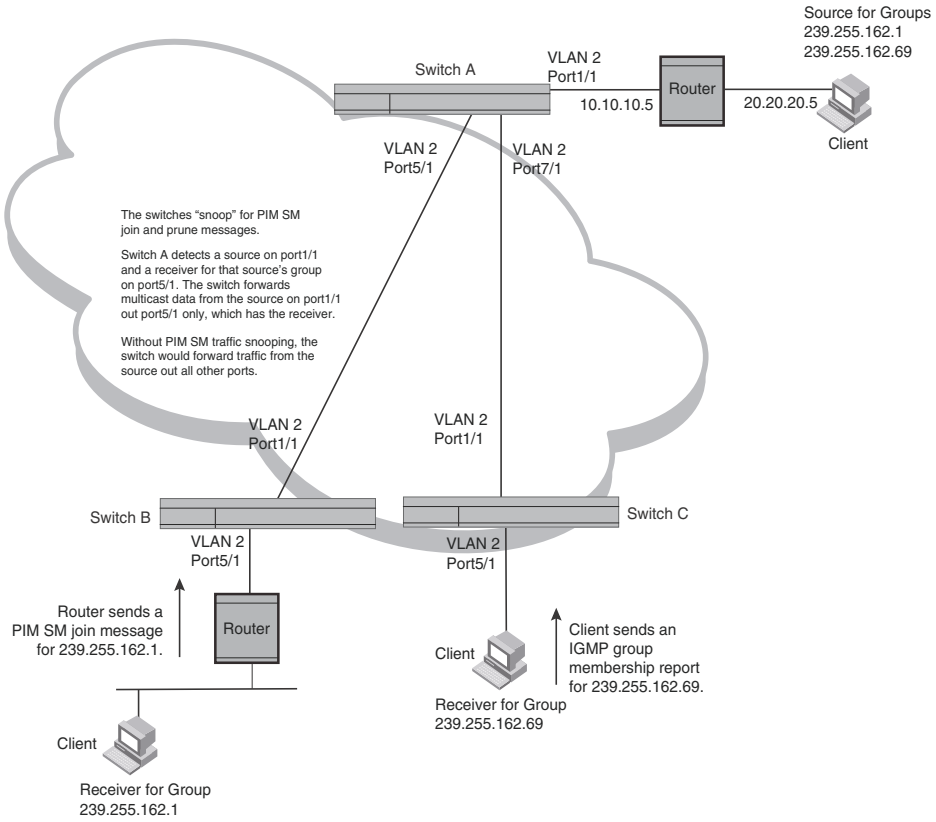
The IP multicast traffic reduction feature and the PIM SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM SM groups learned through join messages as well as MAC addresses learned through IGMP group membership reports. In this case, even though the device never sees a join message for the receiver for group 239.255.162.69, the device nonetheless learns about the receiver and forwards group traffic to the receiver.

The device stops forwarding IP multicast traffic on a port for a group if the port receives a prune message for the group.

Notice that the ports connected to the source and the receivers are all in the same port-based VLAN on the device. This is required for the PIM SM snooping feature. The feature also requires the source and the downstream router to be on different IP subnets, as shown in Figure 164.

Figure 165 shows another example application for PIM SM traffic snooping. This example shows devices on the edge of a Global Ethernet cloud (a Layer 2 Packet over SONET cloud). Assume that each device is attached to numerous other devices such as other PowerConnect's.

FIGURE 165 PIM SM traffic reduction in Global Ethernet environment



The devices on the edge of the Global Ethernet cloud are configured for IP multicast traffic reduction and PIM SM traffic snooping. Although this application uses multiple devices, the feature has the same requirements and works the same way as it does on a single device.

Configuration requirements

Consider the following configuration requirements:

- IP multicast traffic reduction must be enabled on the device that will be running PIM SM snooping. The PIM SM traffic snooping feature requires IP multicast traffic reduction.

  NOTE
  Use the passive mode of IP multicast traffic reduction instead of the active mode. The passive mode assumes that a router is sending group membership queries as well as join and prune messages on behalf of receivers. The active mode configures the device to send group membership queries.

- All the device ports connected to the source and receivers or routers must be in the same port-based VLAN.
- The PIM SM snooping feature assumes that the group source and the device are in different subnets and communicate through a router. The source must be in a different IP subnet than the receivers. A PIM SM router sends PIM join and prune messages on behalf of a multicast group receiver only when the router and the source are in different subnets. When the receiver and source are in the same subnet, they do not need the router in order to find one another. They find one another directly within the subnet.

  The device forwards all IP multicast traffic by default. Once you enable IP multicast traffic reduction and PIM SM traffic snooping, the device initially blocks all PIM SM traffic instead of forwarding it. The device forwards PIM SM traffic to a receiver only when the device receives a join message from the receiver. Consequently, if the source and the downstream router are in the same subnet, and PIM SM traffic snooping is enabled, the device blocks the PIM SM traffic and never starts forwarding the traffic. This is because the device never receives a join message from the downstream router for the group. The downstream router and group find each other without a join message because they are in the same subnet.

NOTE
If the “route-only” feature is enabled on a PowerConnect, PIM SM traffic snooping will not be supported.

Enabling PIM SM traffic snooping

To enable PIM SM traffic snooping, enter the following commands at the global CONFIG level of the CLI.

```
NetIron(config)# ip multicast
NetIron(config)# ip pimsm-snooping
```

The first command enables IP multicast traffic reduction. This feature is similar to PIM SM traffic snooping but listens only for IGMP information, not PIM SM information. You must enable both IP multicast traffic reduction and PIM SM traffic snooping to enable the device to listen for PIM SM join and prune messages.

Syntax: [no] ip multicast [active | passive]

This command enables IP multicast traffic reduction. The **active | passive** parameter specifies the mode. The PIM SM traffic snooping feature assumes that the network has routers that are running PIM SM.

Syntax: [no] ip pimsm-snooping

This command enables PIM SM traffic snooping.

To disable the feature, enter the following command.

```
NetIron(config)# no ip pimsm-snooping
```

If you also want to disable IP multicast traffic reduction, enter the following command.

```
NetIron(config)# no ip multicast
```

Multicast traffic reduction per VLAN or VPLS instance

You can configure the following methods for reducing multicast traffic globally on a PowerConnect router:

- IGMP snooping – This is described in "Changing the IGMP mode" on page 1238.
- PIM snooping – This is described in "PIM SM traffic snooping" on page 1239.

When these are set globally on a router, they apply to all VLANs and all VPLS instances that are configured on the router. You can configure specified VLANs or VPLS instances for multicast traffic reduction by these methods as described in the following sections. Additionally, you are able to configure IGMP and PIM proxy which are only configurable per VLAN or VPLS instance.

Multicast traffic reduction per VPLS instance is supported for dual-mode, untagged, single-tagged, and dual-tagged VPLS endpoints.

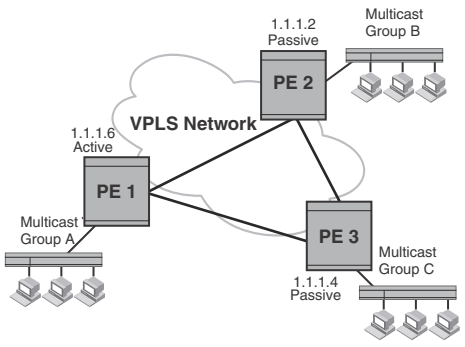
NOTE
IGMP snooping cannot be concurrently enabled on a router with VPLS CPU Protection on a VPLS instance.

NOTE
Traffic will continue to be forwarded only to those VPLS endpoints or peers from which a join for the (S, G) has been recieved irrespective of the status of the local switching option.

Application example

Figure 166 shows an example of multicast traffic reduction in a VPLS network.

FIGURE 166 IP multicast traffic reduction in a VPLS network



In the example shown in Figure 166, when IP multicast traffic reduction (IGMP snooping) is enabled on the VPLS network, PE 1 will be selected as the active port (querier) because it has the lowest router ID amongst the PEs in the VPLS instance. PE 1 will actively send out IGMP queries to solicit information from IP multicast groups within the VPLS instance. PE 2 and PE 3 will not send out queries as they are in passive mode, but will respond to the query from PE 1, with the respective report information received from their hosts.

In a snooping configuration within a VPLS instance, multicast traffic is always flooded to the device that is in active IGMP mode (the router port). If the IP multicast domain includes an IP multicast router, that router will be the querier. In this case, the querier is also known as the router port. If there is no IP multicast router in the domain, one of the devices in the network can be manually configured for active IGMP mode. Otherwise, the device with the lowest router ID will be elected as the querier. In the example in Figure 166, PE 1 is the elected querier.

In a VPLS scenario, reports are always forwarded to all VPLS peers whether or not the peer is the querier. In the above example, PE 1 is elected the querier, but if PE 2 is connected to a receiver, it forwards the reports received from the receiver to both VPLS peers PE 3 and PE 1. Therefore, any traffic for the host connected to PE 2 received from PE 3 are sent by PE 3 to both the router port PE 1 as well as the receiver PE 2.

PE 1 drops the traffic received from PE 3 because it is aware of the presence of a receiver attached to the router PE 2. Traffic sent from PE 3 will only be received by the host connected to PE 2.

If there is no receiver present in the setup, then traffic from PE 3 will only be flooded to its local endpoints and the router port PE 1. Router PE 1 sends the traffic out of its endpoints.

In case of PIM SM snooping, traffic received for unknown groups is always dropped. There is no router port concept in case of PIM SM snooping.

Configuring the IGMP mode per VLAN or VPLS instance

In the following example, multicast traffic reduction is applied using IGMP snooping to VLAN 2.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast passive
```

To remove multicast traffic reduction configurations in VLAN 2, and take the global multicast traffic reduction configuration, enter the following command.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# no multicast
```

In the following example, multicast traffic reduction is applied using IGMP snooping to VPLS instance V1.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls v1 10
NetIron(config--mpls-vpls-v1)# multicast passive
```

Syntax: [no] multicast active | passive

When you enable IP multicast for a specific VLAN or VPLS instance, IGMP snooping is enabled. The device uses IGMP to maintain a table of the Group Membership reports received by the device for the specified VLAN or VPLS instance. You can use active or passive IGMP mode. There is no default mode.

The description for the IGMP modes is as follows:

- **Active** – When active IGMP mode is enabled, the router actively sends out IGMP queries to identify IP multicast groups within the VLAN or VPLS instance and makes entries in the IGMP table based on the Group Membership reports received from the network.
- **Passive** – When passive IGMP mode is enabled, the router listens for IGMP Group Membership reports on the VLAN or VPLS instance specified but does not send IGMP queries. The passive mode is called "IGMP snooping". Use this mode when another device in the VLAN or VPLS instance is actively sending queries.

Configuring the PIM SM traffic snooping per VLAN or VPLS instance

In the following example, multicast traffic reduction is applied using PIM SM Traffic snooping to VLAN 2.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast pimsm-snooping
```

In the following example, multicast traffic reduction is applied using PIM SM traffic snooping to VPLS instance V1.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls v1 10
NetIron(config--mpls-vpls-v1)# multicast pimsm-snooping
```

Syntax: [no] multicast pimsm-snooping

Configuring PIM proxy per VLAN or VPLS instance

Using the PIM proxy function, multicast traffic can be reduced by configuring an PowerConnect router to issue PIM join and prune messages on behalf of hosts that the configured router discovers through standard PIM interfaces. The router is then able to act as a proxy for the discovered hosts and perform PIM tasks upstream of the discovered hosts. Where there are multiple PIM downstream routers, this removes the need to send multiple messages.

To configure a PowerConnect router to function as a PIM proxy on VLAN 2, use the following commands.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast pim-proxy-enable
```

To configure an PowerConnect router to function as an PIM proxy on VPLS instance V1, use the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls v1 10
NetIron(config--mpls-vpls-v1)# multicast pim-proxy-enable
```

Syntax: [no] multicast pim-proxy-enable

Configuring IGMP snooping tracking per VLAN or VPLS instance

When IGMP Snooping Tracking is enabled, the PowerConnect immediately removes any IGMP host port from the IP multicast group entry when it detects an IGMP-leave message on the specified host port without first sending out group-specific queries to the interface. By default, IGMP Snooping Tracking is disabled.

The **ip multicast tracking** command may be enabled globally as well as per VLAN basis. To enable IGMP Snooping Tracking globally, enter a command such as the following.

```
NetIron(config)# multicast tracking
```

Syntax: [no] ip multicast tracking

The **no** form of this command disables the tracking process globally.

To enable IGMP Snooping Tracking per VLAN, enter commands such as the following.

```
NetIron(config)# vlan 100
NetIron(config-vlan-100)# multicast tracking
```

Syntax: [no] multicast tracking

The **no** form of this command disables the tracking process per VLAN.

To enable IGMP Snooping Tracking per VPLS instance, enter commands such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls v1 10
NetIron(config--mpls-vpls-v1)# multicast tracking
```

For IGMPv3, the above command also internally tracks all the IGMPv3 hosts behind a given port. The port is not removed from the IP multicast group entry in the forwarding table until all the hosts behind that port have left that multicast group. When the last IGMPv3 host sends a IGMPv3 leave message, the port is removed from the IP multicast group entry in the forwarding table immediately without first sending out group_source_specific query to the interface

Syntax: [no] multicast tracking

The **no** form of this command disables the tracking process per VPLS instance.

Multicast snooping over VPLS will not load-balance the multicast traffic among multiple tunnels,
Multicast snooping over VPLS will not load-balance the multicast traffic among multiple tunnels,

NOTE
Multicast snooping over VPLS will not load-balance the multicast traffic among multiple tunnels when IGMP Snooping is enabled,

Static IGMP membership

When configuring a static IGMP membership, you have two options:

The **multicast static-group uplink** command which sends the traffic to the router, and saves a port.

The **multicast static-group** <*group-address*> <*port-list*> command is for downstream traffic and uses a port.

Configuring a multicast static group uplink per VLAN

When the **multicast static-group uplink** command is enabled on a snooping VLAN, the snooping device behaves like an IGMP host on ports connected to the multicast router. The snooping device will respond to IGMP queries from the uplink multicast PIM router for the groups and sources configured. Upon the multicast router receiving the IGMP join message, it will initiate the PIM join on its upstream path towards the source to pull the source traffic down. The source traffic will stop at the IGMP snooping device. The traffic will then be forwarded to the multicast receiver and router ports or dropped in hardware if no other multicast receiver and routers are present in the VLAN.

The **multicast static-group uplink** command cannot be configured globally per VPLS basis. It can be configured under the VLAN configuration only.

The **multicast static-group uplink** command must be used with the **multicast static-group** command in order to connect a remote multicast source with the snooping vlan where the static-group is configured.

When using IGMP v3, you can use the **multicast static-group include** or **multicast static-group exclude** command to statically *include* or *exclude* multicast traffic, respectively for hosts that cannot signal group membership dynamically.

To configure the snooping device to statically join a multicast group on the uplink interface, enter commands such as the following.

```
NetIron(config)# vlan 100
NetIron(config-vlan-100)# multicast static-group 224.10.1.1 uplink
```

To configure the physical interface 10.43.3.12 to statically join a multicast group on port 2/4, enter commands such as the following.

```
NetIron(config)# vlan 100
NetIron((config-vlan-100)# multicast static-group 224.10.1.1  2/4
```

To configure the snooping device to statically join a multicast stream with the source address of 10.43.1.12 in the include mode, enter commands such as the following.

```
NetIron(config)# vlan 100
NetIron(config-vlan-100)# multicast static-group 224.10.1.1 include 10.43.1.12
uplink
```

To configure the snooping device to statically join all multicast streams on the uplink interface excluding the stream with source address 10.43.1.12, enter commands such as the following.

```
NetIron(config)# vlan 100
NetIron(config-vlan-100)# multicast static-group 224.10.1.1 exclude 10.43.1.12
uplink
```

Configuring multicast static group <port-list> per VLAN

When the **multicast static-group** <*group-address*> <*port-list*> command is enabled on a snooping VLAN, the snooping device will add the ports to the outgoing interface list of the multicast group entry in the forwarding table as if IGMP joins were received from these ports. These ports will not be aged out from the multicast group for not responding to the IGMP queries.

The **multicast static-group** <*group-address*> <*port-list*> command cannot be configured globally per VPLS basis.

It can be configured under the VLAN configuration level only.

To configure the physical interface ethernet 2/4 to statically join a multicast group, enter commands such as the following.

```
NetIron(config)# vlan 100
NetIron(config-vlan-100)# multicast static-group 224.10.1.1 ethernet 2/4
```

To configure the physical interface ethernet 3/4 to statically join a multicast stream with source address of 10.43.1.12 in the include mode , enter commands such as the following.

```
NetIron(config)# vlan 100
NetIron(config-vlan-100)# multicast static-group 224.10.1.1 include 10.43.1.12
ethernet 3/4
```

To configure the physical interface ethernet 3/4 to statically join all multicast streams on the uplink interface excluding the stream with source address of 10.43.1.12, enter commands such as the following.

```
NetIron(config)# vlan 100
NetIron(config-vlan-100)# multicast static-group 224.10.1.1 exclude 10.43.1.12
ethernet 3/4
```

Syntax: [no] multicast static-group <*group-address*> uplink

Syntax: [no] multicast static-group <*group-address*> <*port-list*>

IGMP v3 Commands

Syntax: [no] multicast static-group <*group-address*> [include I exclude <*source-address*>] uplink

Syntax: [no] multicast static-group <*group-address*> [include I exclude <*source-address*>] <*port-list*>

The **group-address** parameter specifies the group multicast address.

The **include** or **exclude** keyword indicates a filtering action. You can specify which source (for a group) to include or exclude. The **include** or **exclude** keyword is only supported on IGMPv3.

The **source-address** parameter specifies the IP address of the multicast source. Each address must be added or deleted one line per source.

The **uplink** parameter specifies the port as an uplink port that can receive multicast data for the configured multicast groups.. Upstream traffic will be sent to the router and will not use a port.

The **port-list** parameter specifies the range of ports to include in the configuration.

The **no** form of this command removes the static multicast definition. Each configuration must be deleted separately.

Displaying IP multicast information

The following sections show how to display and clear IP multicast reduction information.

Displaying multicast information

You can display IP multicast traffic introduction in a brief mode for all instances or in a detail mode for a specified VLAN or VPLS instance.

```
NetIron#show ip multicast
Global Multicast Traffic Reduction Configuration
   igmp Snooping State:     Active    Version              :          2
   Group Interval     :        260    Query Interval       :        125
   Max Response Time  :         10    Robustness Var       :          1
   Last Member Qry Int:          5    Last Member Qry Count:          3
   Querier Exp Tm     :        255
   igmp Proxy         :   Disabled    Proxy Interval       :         60
   Filter             :   Disabled    Tracking             :   Disabled

   PIM Snooping       :   Disabled
   PIM Prune Wait Time:          3
   PIM Proxy          :   Disabled    Proxy Interval       :         60
VLAN snooping configurations:

----+-----+---------+---------------+-----+-----+-----+--------+-----+-----
VLAN State Mode      Active          PIM   IGMP  PIM   Tracking Group Strm
                     Querier         Snoop Proxy Proxy          Count Count
----+-----+---------+---------------+-----+-----+-----+--------+-----+-----
10   I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
20   I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
30   I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
----+-----+---------+---------------+-----+-----+-----+--------+-----+-----
VPLS State Mode      Active          PIM   IGMP  PIM   Tracking Group Strm
                     Querier         Snoop Proxy Proxy          Count Count
----+-----+---------+---------------+-----+-----+-----+--------+-----+-----
1    I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
2    I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
3    I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
4    I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
5    I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
6    I-Ena Active    Self            I-Dis None  None  I-Dis    0     0
```

Syntax: show ip multicast

To display detailed IP multicast traffic reduction information for a specified VLAN instance on the PowerConnect, enter the following command at any level of the CLI.

```
NetIron#show ip multicast vlan 18
----+-----+---------+--------------+-----+-----+------
VLAN State Mode      Active          Time (*, G)(S, G)
                     Querier         Query Count Count
----+-----+---------+--------------+-----+-----+------
18   Ena   Active    Self            96     3     2
----+-----+---------+--------------+-----+-----+------

Router ports:

Flags: R-Router Port, V2|V3: IGMP Receiver, P_G|P_SG: PIM Join

  1    (*, 229.15.15.1 ) 00:19:15 NumOIF: 3
         Outgoing Interfaces:
              e3/3 vlan 18 ( V2) 00:17:28/22s
              e1/9 vlan 18 ( V2) 00:19:09/21s
              e1/24 vlan 18 ( V2) 00:19:09/23s

  1    (18.0.0.2, 229.15.15.1) in e1/11 vlan 18 00:19:15 NumOIF: 3
         Outgoing Interfaces:
              e3/3 vlan 18 ( V2) 00:17:28/0s
              TR(e1/9,e1/1) vlan 18 ( V2) 00:19:09/0s
              e1/24 vlan 18 ( V2) 00:19:09/0s
         FID: 0x80a8 MVID: None

  2    (*, 229.15.15.7 ) 00:19:16 NumOIF: 3
         Outgoing Interfaces:
              e3/3 vlan 18 ( V2) 00:17:29/23s
              e1/9 vlan 18 ( V2) 00:19:10/22s
              e1/24 vlan 18 ( V2) 00:19:10/24s

  1    (18.0.0.2, 229.15.15.7) in e1/11 vlan 18 00:19:16 NumOIF: 3
         Outgoing Interfaces:
              e3/3 vlan 18 ( V2) 00:17:29/0s
              TR(e1/9,e1/1) vlan 18 ( V2) 00:19:10/0s
              e1/24 vlan 18 ( V2) 00:19:10/0s
         FID: 0x8082 MVID: None

  3    (*, 229.15.15.3 ) 00:19:16 NumOIF: 3
         Outgoing Interfaces:
              e3/3 vlan 18 ( V2) 00:17:29/23s
              e1/9 vlan 18 ( V2) 00:19:10/22s
              e1/24 vlan 18 ( V2) 00:19:10/24s
```

Syntax: show ip multicast vlan <*vlan-id*>

To display detailed IP multicast traffic reduction information for a specified VPLS instance on the PowerConnect, enter the following command at any level of the CLI.

```
NetIron#show ip multicast vpls 1
----+-----+---------+--------------+-----+-----+------
VPLS State Mode      Active          Time (*, G)(S, G)
                     Querier         Query Count Count
----+-----+---------+--------------+-----+-----+------
1    Ena   Active    7.7.7.1         41      1     1
----+-----+---------+--------------+-----+-----+------

Router ports: TNNL peer 7.7.7.1 (14s) VC Label 983040 R Label 1024

Flags: R-Router Port, V2|V3: IGMP Receiver, P_G|P_SG: PIM Join


  1    (*, 229.0.0.1 ) 00:08:01 NumOIF: 1
          Outgoing Interfaces:
               TNNL peer 7.7.7.1 ( R V2) 00:08:01/14s


  1    (192.85.1.31, 229.0.0.1) in e4/8 vlan 1001 00:00:43 NumOIF: 1
          Outgoing Interfaces:
               TNNL peer 7.7.7.1 VC Label 983040 R Label 1024 Port e4/6 ( R V2)
               00:00:43/0s FID: 0x800c MVID:      1
```

Syntax: show ip multicast vpls <*vpls-id*>

You also can display PIM SM information by entering the following command at any level of the CLI

```
NetIron(config)# show ip multicast pimsm-snooping
PIMSM snooping is enabled
VLAN ID 100
  PIMSM neighbor list:
          31.31.31.4 : 12/2 expires 142 s
         31.31.31.13 : 10/7 expires 136 s
          31.31.31.2 : 3/1 expires 172 s
Number of Multicast Groups: 2
1    Group: 239.255.162.4 Num SG 4
     Forwarding ports : 3/1 12/2
     PIMv2 *G join ports : 3/1 12/2
     1     Source: (165.165.165.165, 10/7) FID 0x0bb3
           SG join ports: 12/2 10/7
     2     Source: (161.161.161.161, 10/7) FID 0x0bb2
           SG join ports: 12/2 3/1
     3     Source: (158.158.158.158, 10/7) FID 0x0bb1
           SG join ports: 12/2 3/1
     4     Source: (170.170.170.170, 10/7) FID 0x0baf
           SG join ports: 3/1 10/7
           (S, G) age 0 s
2    Group: 239.255.163.2 Num SG 1
     Forwarding ports : 10/7 12/2
     PIMv2 *G join ports : 10/7 12/2
     1     Source: (165.165.165.165, 3/1) FID 0x0bb5
            SG join ports: 12/2 10/7
```

Syntax: show ip multicast pimsm-snooping

This display shows the following information.

Table 0.15:

This field...	Displays...
The PIM SM traffic snooping state	The first line of the display indicates whether the feature is enabled or disabled; and if it is enabled, if it is passive or active. The PIM SM traffic snooping feature requires the IP multicast traffic reduction feature.
VLAN ID	The port-based VLAN to which the neighbors and groups listed below the VLAN ID apply. Each port-based VLAN is a separate Layer 2 broadcast domain. **NOTE:** PIM SM traffic snooping requires the source and the receivers to be in the same port-based VLAN on the device. If the source and receivers are in different port-based VLANs, the device blocks the multicast traffic.
PIM SM Neighbor list	The PIM SM routers that are attached to the device's ports in the VLAN. The value following "expires" indicates how many seconds the device will wait for a hello message from the neighbor before determining that the neighbor is no longer present and removing the neighbor from the list.
Number of Multicast Group	The total number of groups for which the VLAN's ports have received PIM join or prune messages and IGMP group membership reports.
Multicast Group	The IP address of the multicast group. The "Num SG" entry indicates how many Source to Group flows are created for that Multicast Group as there can be more than one source for a given group. **NOTE:** The fid and camindex values are used by Dell Technical Support for troubleshooting.
Forwarding Port	The ports attached to the group's receivers. A port is listed here when it receives a join message for the group, an IGMP membership report for the group, or both.
PIMv2 Group Port	The ports on which the PowerConnect has received PIM SM join messages for the group.
Source, Port list	The IP address of each PIM SM source and the PowerConnect ports connected to the receivers of the source.
SG join ports:	Ports from which a join message was received. The PowerConnect forwards the traffic only on this port.
(S, G) age	The actual aging value. If this entry shows the value 0 seconds, software age value is still 0 and the flow is programmed in the CAM. If the entry shows a value other than 0 seconds, then the CAM entry has aged out and the software aging has begun. Once this age value reaches the Group Age value the entry will be deleted from the table. Group age value can be can be from 10 – 1220 seconds. The default is 260 seconds.

Displaying IP multicast statistics

To display IP multicast statistics on a device, enter the following commands at any level of the CLI.

```
NetIron# show ip multicast statistics
IP multicast is enabled - Passive

VLAN ID 1
Reports Received:               34
Leaves Received:                21
```

```
General Queries Received:        60
Group Specific Queries Received: 2
Others Received:                 0
General Queries Sent:            0
Group Specific Queries Sent:     0

VLAN ID 2
Reports Received:                0
Leaves Received:                 0
General Queries Received:        60
Group Specific Queries Received: 2
Others Received:                 0
General Queries Sent:            0
Group Specific Queries Sent:     0
```

The command in this example shows statistics for two port-based VLANs.

Syntax: show ip multicast statistics

Clearing IP multicast statistics

To clear IP multicast statistics on a device, enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# clear ip multicast statistics
```

This command resets statistics counters for all the statistics displayed by the **show ip multicast statistics** command to zero.

Syntax: clear ip multicast statistics

Clearing IGMP group flows

To clear all the IGMP flows learned by the device, enter the following command at the Privileged EXEC level of the CLI.

```
NetIron# clear ip multicast all
```

The following example shows IGMP flows information listed by the **show ip multicast** command, followed by removal of the information by the **clear ip multicast all** command.

```
NetIron# show ip multicast
IP multicast is enabled - Active
VLAN ID 1
Active 192.168.2.30 Router Ports  4/13
Multicast Group: 239.255.162.5, Port: 4/4 4/13
Multicast Group: 239.255.162.4, Port: 4/10 4/13

NetIron# clear ip multicast all

NetIron# show ip multicast
IP multicast is enabled - Active
VLAN ID 1
Active 192.168.2.30 Router Ports  4/13
```

To clear the learned IGMP flows for a specific IP multicast group, enter a command such as the following.

```
NetIron# clear ip multicast group 239.255.162.5
```

The following example shows how to clear the IGMP flows for a specific group and retain reports for other groups.

```
NetIron# show ip multicast
IP multicast is enabled - Active
VLAN ID 1
Active 192.168.2.30 Router Ports  4/13
Multicast Group: 239.255.162.5, Port: 4/4 4/13
Multicast Group: 239.255.162.4, Port: 4/10 4/13

NetIron# clear ip multicast group 239.255.162.5

NetIron# show ip multicast
IP multicast is enabled - Active
VLAN ID 1
Active 192.168.2.30 Router Ports  4/13
Multicast Group: 239.255.162.4, Port: 4/10 4/13
```

Syntax: clear ip multicast all | group <*group-id*>

The **all** parameter clears the learned flows for all groups.

The **group** <*group-id*> parameter clears the flows for the specified group but does not clear the flows for other groups.

Configuring MBGP 28

Overview

The following displays the Multi-protocol Border Gateway Protocol (MBGP) features supported by PowerConnect B-MLXe.

- MBGP
- Advertising Routes from the Local AS to MBGP
- Network Prefix to Advertise
- Redistribution of Directly-Connected Multicast Routes into MBGP
- Static IP Multicast Routes
- Aggregating Routes Advertised to BGP4 Neighbors
- Displaying MBGP Information
- Clearing MBGP Information
- IPv6 Support
- VRF Support
- IPv4 Multicast
- IPv6 Multicast

This chapter provides details on how to configure ***Multi-protocol Border Gateway Protocol (MBGP)***. MBGP is an extension to BGP that allows a router to support separate unicast and multicast topologies. BGP4 cannot support a multicast network topology that differs from the network's unicast topology. MBGP allows you to support a multicast topology that is distinct from the network's unicast topology. For example, if you want to dedicate a link on your Internet router to multicast traffic, use MBGP to handle the routes on that link.

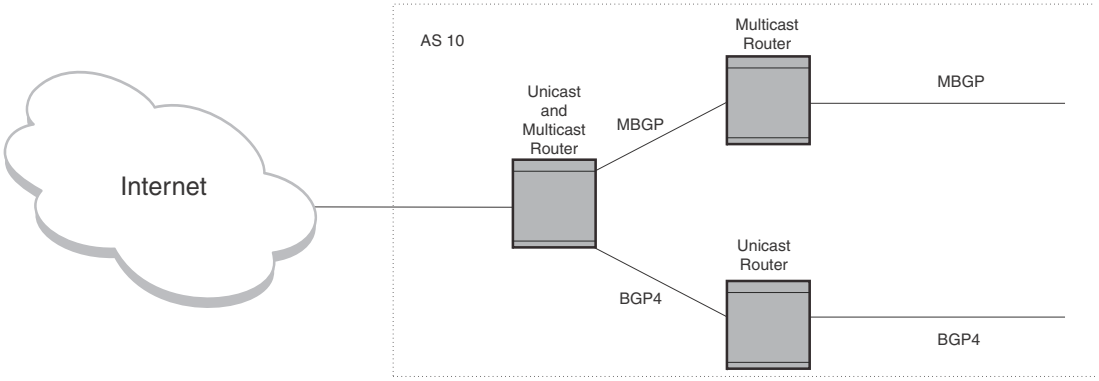
NOTE
The IPv6 multicast address family for MBGP is supported. However, only static mroutes and directly connected routes over IPv6 multicast enabled interfaces may be enabled for distribution.

MBGP provides the following benefits:

- You can support a network whose multicast topology is different from its unicast topology. Even if the unicast and multicast networks have the same topologies, you can support different sets of routing policies for unicast and multicast.
- You can use BGP4's powerful feature set with MBGP.

Figure 167 shows an example of a network that contains both a unicast topology and a multicast topology. The unicast and multicast router in this example receives unicast and multicast routes from the Internet. The router advertises the multicast routes to the multicast router and advertises the unicast routes to the unicast router. Likewise, the unicast and multicast router can advertise unicast routes received from the unicast router to the Internet, and can advertise multicast routes received from the multicast router to the Internet.

FIGURE 167 MBGP used when multicast topology is different from unicast topology



An MBGP router learns MBGP routes from its neighbors in other ASs. An MBGP router also can advertise MBGP routes to its neighbors. The implementation of MBGP enables you to advertise multicast routes from the following sources:

- Explicitly configured network prefixes
- Static IP multicast routes
- Directly-connected multicast routes redistributed into MBGP.

You can configure an aggregate address to aggregate network prefixes into a single, more general prefix for advertisement.

MBGP is described in detail in RFC 2858.

Configuration considerations

The configuration considerations are as follows:

- MBGP does not redistribute DVMRP routes. It redistributes static routes only.
- You cannot redistribute MBGP routes into BGP4.
- By default, the PowerConnect does not place a limit any limit on the number of multicast routes. You can configure the device to place a limit on the number of multicast routes by using the **ip max-mroute** command.

Configuring MBGP

1. Optional – Set the maximum number of multicast routes supported by the PowerConnect.
2. Enable MBGP by doing the following:

- Enable PIM Sparse Mode (PIM SM) or PIM Dense Mode (PIM DM) globally and on the individual Reverse Path Forwarding (RPF) interfaces. PIM must be running on the PowerConnect in order for the device to send multicast prefixes to other multicast devices.
- Enable BGP4. If this is the first time you have configured BGP4 on this device, you also need to specify the local AS number.

3. Identify the neighboring MBGP routers.
4. Optional – Configure an MBGP default route.
5. Optional – Configure an IP multicast static route.
6. Optional – Configure an MBGP aggregate address.
7. Optional – Configure a route map to apply routing policy to multicast routes.
8. Save the configuration changes to the startup-config file.

Setting the maximum number of multicast routes supported

NOTE
This procedure requires a software reload to place the change into effect.

Use the **system-max multicast-route** command to increase the maximum number of multicast routes for IPv6 address families.

Entering the following commands.

```
NetIron(config)# system-max multicast-route 12000
NetIron(config)# write memory
NetIron(config)# end
NetIron# reload
```

These commands increase the maximum number of multicast routes supported, save the configuration change to the startup-config file, and reload the software to place the change into effect.

Syntax: [no] system-max multicast-route <*num*>

The <*num*> parameter specifies the number of multicast routes. This value can range from 1024 to the maximum routes supported by the system, subject to a maximum of 153,600 routes.

You can use the following runtime command to define the maximum number of multicast routes supported. This parameter can be defined for the default VRF using the following command.

```
NetIron(config)# ip max-mroute
```

Syntax: [no] ip max-mroute <*num*>

The <*num*> parameter specifies the number of multicast routes and can be from 1024 – 153,600.

To define the maximum number of multicast routes for a specified VRF, use the following commands.

```
NetIron(config)# ip vrf blue
NetIron(config-vrf-blue)# ip max-mroute
```

Syntax: [no] ip vrf <*vrf-name*>

Syntax: [no] ip max-mroute <*num*>

The **ip vrf** parameter specifies the virtual routing instance (VRF) specified by the variable *<vrf-name>*

The *<num>* parameter specifies the number of multicast routes. This value can range from 1024 to the maximum routes supported by the system, subject to a maximum of 153,600 routes.

Enabling MBGP

To make use of MBGP4, you must enable PIM SM or DM and BGP4. Enter commands such as the following.

```
NetIron> enable
NetIron# configure terminal
NetIron(config)# router pim
NetIron(config)# interface ethernet 1/1
NetIron(config-if-1/1)# ip address 1.1.1.1/24
NetIron(config-if-1/1)# ip pim
NetIron(config-if-1/1)# exit
NetIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
NetIron(config-bgp)# local-as 10
```

NOTE
For IPv6 address family, make sure you enter the IPv6 versions of the commands.

The commands in this example configure PIM DM globally and on port 1/1, then enable BGP4. Once you enable PIM DM or PIM SM both globally and on the individual RPF interfaces, and enable BGP4, support for MBGP is automatically enabled.

Once MBGP is enabled, MBGP parameters are configured under the IPv4 multicast address family. Enter the following command to enter the IPv4 multicast address family level.

```
NetIron(config-bgp)#address-family ipv4 multicast
NetIron(config-bgp-ipv4m)#
```

Syntax: [no] address-family ipv4 multicast

To enable MBGP for IPv6, enter the following command.

```
NetIron(config-bgp)#address-family ipv6 multicast
NetIron(config-bgp-ipv6m)#
```

Syntax: [no] address-family ipv6 multicast

Adding MBGP neighbors

To add an MBGP neighbor, enter a command such as the following.

```
NetIron(config-bgp-ipv4m)#neighbor 1.2.3.4 remote-as 44
```

This command adds a router with IP address 1.2.3.4 as an MBGP neighbor.

The **remote-as 44** parameter specifies that the neighbor is in remote BGP4 AS 44. The PowerConnect will exchange only multicast routes with the neighbor.

NOTE
If the PowerConnect has multiple neighbors with similar attributes, you can simplify configuration by configuring a peer group, then adding individual neighbors to it. The configuration steps are similar, except you specify a peer group name instead of a neighbor IP address when configuring the neighbor parameters, then add individual neighbors to the peer group.

The command is the same as the command for configuring a unicast BGP neighbor, except in MBGP, the command is entered in the IPv4 multicast address family level. Here is the full syntax for the neighbor command.

Syntax: **[no] neighbor** <*ip-addr*> **|** <*peer-group-name*>
[default-originate [route-map <*map-name*>**]]**
[description <*string*>**]**
[distribute-list in | out <*num,num,...*> **|** <*acl-num*> **in | out]**
[ebgp-multihop [<*num*>**]]**
[filter-list in | out <*num,num,...*> **|** <*acl-num*> **in | out | weight]**
[maximum-prefix <*num*> *[*<*threshold*>**] [teardown]]**
[next-hop-self]
[password [0 | 1] <*string*>**]**
[prefix-list <*string*> **in | out]**
[remote-as <*as-number*>**]**
[remove-private-as]
[route-map in | out <*map-name*>**]**
[route-reflector-client]
[send-community]
[soft-reconfiguration inbound]
[shutdown [generate-rib-out]]
[timers keep-alive <*num*> **hold-time** <*num*>**]**
[update-source loopback <*num*>**]**
[weight <*num*>**]**

The <*ip-addr*> | <*peer-group-name*> parameter indicates whether you are configuring an individual neighbor or a peer group. If you specify a neighbor's IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group. Make sure you enter the IP address in the correct address family format.

The **remote-as** <*as-number*> parameter specifies the AS the MBGP neighbor is in. The <*as-number*> can be a number from 1 – 65535. There is no default.

NOTE
The PowerConnect attempts to establish a BGP4 session with a neighbor as soon as you enter a command specifying the neighbor's IP address. If you want to completely configure the neighbor parameters before the PowerConnect establishes a session with the neighbor, you can administratively shut down the neighbor.

Optional configuration tasks

The following sections describe how to perform some optional BGP4 configuration tasks.

NOTE
This section shows some of the more common optional tasks, including all the tasks that require you to specify that they are for MBGP. Most tasks are configured only for BGP4 but apply both to BGP4 and MBGP.

Advertising routes from the local AS to MBGP

You can configure the PowerConnect to advertise directly-connected and static multicast routes from the local AS to other ASs using the following methods:

- **For directly-connected routes:**
  - Enable redistribution of directly-connected multicast routes.
- **For indirectly-connected routes:**
  - Configure static IP multicast routes. The corresponding IP route must be present in the IP multicast table.
  - Explicitly configure network prefixes to advertise (**network** command).

NOTE
You can configure the device to advertise directly-connected networks into MBGP using the **network** command. You are not required to use redistribution or configure static multicast routes.

Configuring a network prefix to advertise

By default, the PowerConnect advertises MBGP routes only for the networks you identify using the network command or that are redistributed into MBGP from IP multicast route tables.

NOTE
The exact route must exist in the IP multicast route table so that the PowerConnect can create a local MBGP route.

To configure the PowerConnect to advertise network 207.95.22.0/24 as a multicast route, enter the following command.

```
NetIron(config-bgp-ipv4m)# network 207.95.22.0 255.255.255.0
```

Syntax: [no] network <*ip-addr*> <*ip-mask*> **[route-map** <*map-name*>**] [backdoor] [weight** <*num*>**]**

The <*ip-addr*> is the network number and the <*ip-mask*> specifies the network mask.

NOTE
For IPv6 address family, make sure you enter the IP address in IPv6 format.

The **route-map** <*map-name*> parameter specifies the name of the route map you want to use to set or change BGP4 attributes for the network you are advertising. The route map must already be configured.

The **backdoor** parameter changes the administrative distance of the route to this network from the EBGP administrative distance (20 by default) to the Local BGP weight (200 by default), thus tagging the route as a backdoor route.

The **weight** <*num*> parameter specifies a weight to be added to routes to this network.

Enabling redistribution of directly-connected multicast routes into MBGP

To redistribute a directly-connected multicast route into MBGP enable redistribution of directly-connected routes into MBGP, using a route map to specify the routes to be redistributed.

Example

```
NetIron(config)# access-list 10 permit 207.95.22.0 0.0.0.255
NetIron(config)# route-map mbgpmap permit 1
NetIron(config-routemap mbgpmap)# match ip address 10
NetIron(config-routemap mbgpmap)# exit
NetIron(config)# router bgp
NetIron(config-bgp-ipv4m)# redistribute connected route-map mbgpmap
```

The first command configures an IP ACL for use in the route map. The ACL matches on the destination network for the route to be redistributed. The next four commands configure a route map that matches on routes to the multicast network specified in IP ACL 10. The PowerConnect redistributes routes that match the route map into MBGP.

Syntax: [no] **redistribute [connected | static] [metric** <*num*>**] [route-map** <*map-name*>**]**

The **connected** parameter indicates that you are redistributing routes to directly attached devices into MBGP.

The **static** parameter indicates that you are redistributing static mroutes into MBGP.

The **metric** <*num*> parameter changes the metric. You can specify a value from 0 – 4294967295. The default
is 0.

The **route-map** <*map-name*> parameter specifies a route map to be consulted before redistributing the routes into MBGP.

NOTE
The route map you specify must already be configured.

NOTE
For IPv6 address family, make sure you enter the IPv6 versions of the commands.

Configuring static IP multicast routes

To configure static IP multicast routes, enter commands such as the following.

```
NetIron(config)# ip mroute 207.95.10.0 255.255.255.0 interface ethernet 1/2
NetIron(config)# ip mroute 0.0.0.0 0.0.0.0 interface ethernet 2/3
```

The commands in this example configure two static multicast routes. The first route is for a specific source network, 207.95.10.0/24. If the PowerConnect receives multicast traffic for network 207.95.10.0/24, the traffic must arrive on port 1/2. The second route is for all other multicast traffic. Traffic from multicast sources other than 207.95.10.0/24 must arrive on port 2/3.

If you configure more than one static multicast route, the PowerConnect always uses the most specific route that matches a multicast source address. Thus, if you want to configure a multicast static route for a specific multicast source and also configure another multicast static route for all other sources, you can configure two static routes as shown in this example.

Syntax: [no] ip mroute <*ip-addr*> <*ip-mask*> [<*next-hop-ip-addr*> | ethernet <*slot/port*> | ve <*num*> | tunnel <*num*> | null0] [<*cost*>] [distance < *num*>]

The **ip-addr** and **ip-mask** parameters specifies the PIM source for the route. Also, for IPv6 address family, make sure you enter the IP address in IPv6 format.

The **ethernet** <*slot/port*> parameter specifies a physical port.

The **ve** <*num*> parameter specifies a virtual interface.

The **tunnel** <*num*> parameter specifies a GRE tunnel interface that is being configured. The GRE tunnel interface is enabled under the router PIM configuration.

The **null0** parameter is the same as dropping the traffic.

The **distance** <*num*> parameter sets the administrative distance for the route.

The <*cost*> parameter specifies the cost metric of the route.
Possible values are: 1 - 6
Default value: 1

NOTE
Regardless of the administrative distances, the PowerConnect always prefers directly connected routes over other routes.

Aggregating routes advertised to BGP4 neighbors

By default, the PowerConnect advertises individual MBGP routes for all the multicast networks. The aggregation feature allows you to configure the PowerConnect to aggregate routes in a range of networks into a single CIDR number. For example, without aggregation, the PowerConnect will individually advertise routes for networks 207.95.10.0/24, 207.95.20.0/24, and 207.95.30.0/24. You can configure the PowerConnect to instead send a single, aggregate route for the networks. The aggregate route would be advertised as 207.95.0.0/16.

To aggregate MBGP routes for 207.95.10.0/24, 207.95.20.0/24, and 207.95.30.0/24, enter the following command.

```
NetIron(config-bgp-router)# aggregate-address 207.95.0.0 255.255.0.0
```

Syntax: [no] aggregate-address <*ip-addr*> <*ip-mask*> [as-set] [summary-only] [suppress-map <*map-name*>] [advertise-map <*map-name*>] [attribute-map <*map-name*>]

The <*ip-addr*> and <*ip-mask*> parameters specify the aggregate value for the networks. Also, for IPv6 address family, make sure you enter the IP address in IPv6 format.

The **as-set** parameter causes the router to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.

The **summary-only** parameter prevents the router from advertising more specific routes contained within the aggregate route.

The **suppress-map** <*map-name*> parameter prevents the more specific routes contained in the specified route map from being advertised.

The **advertise-map** <*map-name*> parameter configures the PowerConnect to advertise the more specific routes in the specified route map.

The **attribute-map** <*map-name*> parameter configures the PowerConnect to set attributes for the aggregate routes based on the specified route map.

NOTE
For the **suppress-map**, **advertise-map**, and **attribute-map** parameters, the route map must already be defined.

Displaying MBGP information

All of the BGP show commands have MBGP equivalents. Use **mbgp** instead of **bgp** in the command syntax. For example, to display the MBGP route table, enter the **show ip mbgp routes** command instead of the **show ip bgp routes** command.

Table 195 lists the MBGP show commands and describes their output. For information about a command, refer to 26, "Configuring BGP4 (IPv4)".

TABLE 195 MBGP show commands for IPv4

Command	Description
show ip mbgp summary	Displays summary configuration information and statistics.
show ip mbgp config	Shows the configuration commands in the running-config.
show ip mbgp neighbors	Displays information about MBGP neighbors.
show ip mbgp peer-group	Displays information about MBGP peer groups.
show ip mbgp routes	Displays MBGP routes.
show ip mbgp <*ip-addr*>[/<*prefix*>]	Displays a specific MBGP route.
show ip mbgp attribute-entries	Displays MBGP route attributes.
show ip mbgp dampened-paths	Displays MBGP paths that have been dampened by route flap dampening.
show ip mbgp flap-statistics	Displays route flap dampening statistics.
show ip mbgp filtered-routes	Displays routes that have been filtered out.
show ip mbgp vpn4	Displays VPN-IPv4 address family information
show ip mbgp vrf	Displays IPv4 address family information for a VPN Routing/Forwarding instance

Table 196 lists the show commands available to display MBGP IPv6 information:

TABLE 196 MBGP show commands for IPv6

Command	Description
show ipv6 mbgp summary	Displays summary configuration information and statistics.
show ipv6 mbgp config	Shows the configuration commands in the running-config.
show ipv6 mbgp neighbors	Displays information about MBGP neighbors.
show ip mbgp peer-group	Displays information about MBGP peer groups.
show ipv6 mbgp routes	Displays MBGP routes.
show ipv6 mbgp <*ip-addr*>[/<*prefix*>]	Displays a specific MBGP route.
show ipv6 mbgp attribute-entries	Displays MBGP route attributes.
show ipv6 mbgp dampened-paths	Displays MBGP paths that have been dampened by route flap dampening.

TABLE 196 MBGP show commands for IPv6 (Continued)

Command	Description
show ipv6 mbgp flap-statistics	Displays route flap dampening statistics.
show ipv6 mbgp filtered-routes	Displays routes that have been filtered out.

The following sections show examples of some of the MBGP show commands. An example of the **show ip mroute** and the **show ipv6 mroute** commands are also included. Both of the commands display the multicast route table.

Displaying summary MBGP information

To display a summary of MBGP IPv4 information, enter the following command at any CLI prompt.

```
NetIron# show ip mbgp summary
   BGP4 Summary
  Router ID: 9.9.9.1   Local AS Number : 200
  Confederation Identifier : not configured
  Confederation Peers:
  Maximum Number of Paths Supported for Load Sharing : 1
  Number of Neighbors Configured : 1, UP: 1
  Number of Routes Installed : 5677
  Number of Routes Advertising to All Neighbors : 5673
  Number of Attribute Entries Installed : 3
  Neighbor Address  AS#   State   Time     Rt:Accepted Filtered Sent   ToSend
  166.1.1.2         200   ESTAB   0h24m54s    3        0        5673   0
```

Syntax: show ip mbgp summary

NOTE
This command's display looks similar to the display for the **show ip bgp config** command. However, the **show ip mbgp config** command lists only the MBGP neighbors, whereas the show ip bgp config command lists only the BGP neighbors.

To display a summary of MBGP IPv6 information, enter the following command at any CLI prompt.

```
NetIron# show ipv6 mbgp summary
  BGP4 Summary
  Router ID: 1.1.1.1   Local AS Number: 100
  Confederation Identifier: not configured
  Confederation Peers:
  Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
  Number of Neighbors Configured: 1, UP: 1
  Number of Routes Installed: 4, Uses 344 bytes
  Number of Routes Advertising to All Neighbors: 7, Uses 308 bytes
  Number of Attribute Entries Installed: 2, Uses 184 bytes
  Neighbor Address  AS#   State   Time     Rt:Accepted Filtered Sent   ToSend
  2004::2           200   ESTAB   0h39m50s    4        0         3      0
```

Syntax: show ipv6 mbgp summary

Displaying the active MBGP configuration

To display the active MBGP IPv4 configuration information contained in the running-config without displaying the entire running-config, enter the following command at any level of the CLI.

```
NetIron# show ip mbgp config
Current BGP configuration:

router bgp
 local-as 200
 neighbor 166.1.1.2 remote-as 200

 address-family ipv4 unicast
 no neighbor 166.1.1.2 activate
 exit-address-family

 address-family ipv4 multicast
 redistribute connected
 redistribute static
 neighbor 166.1.1.2 activate
 exit-address-family

 address-family ipv6 unicast
 exit-address-family
end of BGP configuration
```

Syntax: show ip mbgp config

NOTE
This command displays exactly the same information as the **show ip bgp config** command. Each command displays both the BGP and MBGP configuration commands that are in the running-config.

To display the active MBGP IPv6 configuration information contained in the running-config without displaying the entire running-config, enter the following command at any level of the CLI.

```
NetIron# show ipv6 mbgp config
Current BGP configuration:

router bgp
 local-as 100
 neighbor 2004::2 remote-as 200

 address-family ipv4 unicast
 no neighbor 2004::2 activate
 exit-address-family

 address-family ipv4 multicast
 exit-address-family

 address-family ipv6 unicast
 exit-address-family

 address-family ipv6 multicast
 neighbor 2004::2 activate
 redistribute connected
 redistribute static
 exit-address-family

 address-family vpnv4 unicast
 exit-address-family
end of BGP configuration
```

Syntax: show ipv6 mbgp config

Displaying MBGP neighbors

To view MBGP IPv4 neighbor information including the values for all the configured parameters, enter the **show ip mbgp neighbor** command. This display is similar to the **show ip bgp neighbor** display but has additional fields that apply only to MBGP. These fields are shown in bold type in the example and are explained below.

NOTE
The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

```
NetIron # show ip mbgp neighbor 7.7.7.2
    Total number of BGP Neighbors: 1
1   IP Address: 166.1.1.2, Remote AS: 200 (IBGP), RouterID: 8.8.8.1
    State: ESTABLISHED, Time: 0h33m26s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 9 seconds, HoldTimer Expire in 161 seconds
       PeerGroup: mbgp-mesh
       MD5 Password: $Gsig@U\
       NextHopSelf: yes
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 2       3264    17        0            0
       Received: 1       1       34        0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  multicast  capability
      Peer configured for  IPV4 multicast  Routes
    TCP Connection state: ESTABLISHED, MD5-Password: ********
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:   284418, Received: 767
       Local host:  166.1.1.1, Local  Port: 179
       Remote host: 166.1.1.2, Remote Port: 8137
       ISentSeq:    2763573  SendNext:    3047992  TotUnAck:          0
       TotSent:      284419  ReTrans:           0  UnAckSeq:    3047992
       IRcvSeq:     3433336  RcvNext:     3434104  SendWnd:       65000
        TotalRcv:        768  DupliRcv:          0  RcvWnd:        65000
       SendQue:           0  RcvQue:            0  CngstWnd:       1440
```

Syntax: show ip mbgp neighbors [<*ip-addr*>]

The <*ip-addr*> parameter specifies the neighbor's IP address.

To view MBGP IPv6 neighbor information including the values for all the configured parameters, enter the following command.

NOTE
The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

```
NetIron # show ipv6 mbgp neighbor 2004::2
1   IP Address: 2004::2, AS: 200 (EBGP), RouterID: 2.2.2.2, VRF: default-vrf
    State: ESTABLISHED, Time: 0h47m45s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 19 seconds, HoldTimer Expire in 163 seconds
    Minimal Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 7       4       201       6            0
       Received: 7       3       207       0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx:  0h49m43s    ---           Rx: 0h47m45s   ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV6  multicast  capability
      Peer configured for IPV6  multicast  Routes
    TCP Connection state: ESTABLISHED, flags:00000044 (0,0)
    TTL check: 0, value: 0, rcvd: 0
       Byte Sent:   1071, Received: 1291
       Local host:  2004::1, Local  Port: 179
       Remote host: 2004::2, Remote Port: 8202
       ISentSeq:  679044370  SendNext:  679045442  TotUnAck:          0
       TotSent:        1072  ReTrans:           0  UnAckSeq:  679045442
       IRcvSeq:   678124443  RcvNext:   678125735  SendWnd:       65000
       TotalRcv:       1292  DupliRcv:          0  RcvWnd:        65000
       SendQue:           0  RcvQue:            0  CngstWnd:       1440
```

Syntax: show ipv6 mbgp neighbors [<*ipv6-addr*>]

The <*ipv6-addr*> parameter specifies the neighbor's IPv6 address.

Both examples show how to display information for a specific neighbor, by specifying the neighbor's IP address with the command. The number in the far left column, for example 1, on the first line following the issued command, is similar to an index that indicates the neighbor for which the information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The Neighbor NLRI Negotiation section (shown in bold type) lists the types of routes that this PowerConnect can exchange with the MBGP neighbor.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the PowerConnect's Transmission Control Block (TCB) for the TCP session between the PowerConnect and its neighbor. These fields are described in detail in section 3.2 of RFC 793, "Transmission Control Protocol Functional Specification".

Displaying MBGP routes

To display the MBGP IPv4 route table, enter the following command.

```
NetIron#show ip mbgp route
Total number of BGP Routes: 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED s:STALE
       Prefix           Next Hop        Metric     LocPrf     Weight  Status
1      8.8.8.0/24       166.1.1.2       0          100        0       BI
       AS_PATH:
2     31.1.1.0/24       166.1.1.2       0          100        0       BI
       AS_PATH:
```

Syntax: show ip mbgp routes

To display the MBGP IPv6 route table, enter the following command.

```
NetIron#show ipv6 mbgp route
Total number of BGP Routes: 4
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m: NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      2003::/64          2004::2         0          100        0      BE
         AS_PATH: 200
2      2004::/64          2004::2         0          100        0      BE
         AS_PATH: 200
3      2005::/64          2004::2         0          100        0      BE
         AS_PATH: 200
4      2017::1/128        2004::2                    100        0      BE
         AS_PATH: 200 700
```

Syntax: show ipv6 mbgp routes

Displaying the IP Multicast Route Table

To display the IPv4 multicast route table, enter the following command.

```
NetIron#show ip mroute
 Type Codes - B:BGP  D:Connected  S:Static;   Cost - Dist/Metric
        Destination        Gateway         Port        Cost       Type
1       9.9.9.0/30         DIRECT          loopback 1  0/0        D
2       20.1.1.0/24        DIRECT          ve 220      0/0        D
3       101.1.1.0/24       DIRECT          ve 1        0/0        D
4       101.1.2.0/24       DIRECT          ve 2        0/0        D
5       101.1.3.0/24       DIRECT          ve 3        0/0        D
6       101.1.4.0/24       DIRECT          ve 4        0/0        D
7       101.1.5.0/24       DIRECT          ve 5        0/0        D
8       101.1.6.0/24       DIRECT          ve 6        0/0        D
9       101.1.7.0/24       DIRECT          ve 7        0/0        D
10      101.1.8.0/24       DIRECT          ve 8        0/0        D
11      8.8.8.0/24         166.1.1.2       eth 4/1     200/0      B
12      31.1.1.0/24        166.1.1.2       eth 4/1     200/0      B
```

Syntax: show ip mroute [*<ip-addr> <ip-mask>* **| bgp | static**]

The <*ip-addr*> <*ip-mask*> options display IPv4 multicast route information for a specific destination address only.

The **bgp** parameter displays IPv4 multicast route information for BGP routes only.

The **static** parameter displays IPv4 multicast route information for static routes only.

To display the IPv6 multicast route table, enter the following command.

```
NetIron#show ipv6 mroute
 IPv6 Routing Table - 6 entries:
Type Codes -  B:BGP C: Connected I:ISIS L:Local O:OSPF R:RIP S:Static
OSPF  Type:  i - Inter, 1 - External Type1, 2 - External Type2, e - External
Type IPv6 Prefix               Next Hop Router           Interface  Dis/Metric
B  2003::/64                   2004::2                   eth 4/4     20/1
C  2004::/64                   ::                        eth 4/4     0/0
B  2005::/64                   2004::2                   eth 4/4     20/1
C  2007::/64                   ::                        eth 4/2     0/0
C  2010::1/128                 ::                        loopback 1  0/0
B  2017::1/128                 2004::2                   eth 4/4     20/2
```

Syntax: show ipv6 mroute [<*ip-addr*> <*ip-mask*> **| bgp | static**]

The <*ip-addr*> <*ip-mask*> options display IPv6 multicast route information for a specific destination address only.

The **bgp** parameter displays IPv6 multicast route information for BGP routes only.

The **static** parameter displays IPv6 multicast route information for static routes only.

Displaying MBGP Attribute Entries

To display MBGP Attributes for IPv4.

```
NetIron#show ip mbgp attribute-entries
        Total number of BGP Attribute Entries: 4
1       Next Hop  :172.4.3.7          Metric   :0              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :700 (length 3)
        Address: 0x27ace0c4  Hash:241 (0x0300067a)
        Links: 0x00000000, 0x00000000, nlri: 0x27b4e874
        Reference Counts: 3:0:0, Magic: 19
2       Next Hop  :172.4.3.7          Metric   :1              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :700 (length 3)
        Address: 0x27ace184  Hash:242 (0x0300067a)
        Links: 0x00000000, 0x00000000, nlri: 0x27b4e8ce
        Reference Counts: 1:0:0, Magic: 20
3       Next Hop  :172.4.4.1          Metric   :300            Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :100 (length 3)
        Address: 0x27ace064  Hash:615 (0x030001ca)
        Links: 0x00000000, 0x00000000, nlri: 0x27b4e27a
        Reference Counts: 3:0:0, Magic: 18
```

Syntax: show ip mbgp attribute-entries

To display MBPG attributes for IPv6, enter the following command.

```
NetIron#show ipv6 mbgp attribute-entries
        Total number of BGP Attribute Entries: 10
1       Next Hop  :2004::1            Metric   :100             Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :100 (length 3)
        Address: 0x27c9643c  Hash:415 (0x030001ca)
        Links: 0x00000000, 0x00000000, nlri: 0x00000000
        Reference Counts: 0:0:3, Magic: 16
2       Next Hop  :2003::7            Metric   :0               Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :700 (length 3)
        Address: 0x27c961b4  Hash:496 (0x0300067a)
        Links: 0x00000000, 0x00000000, nlri: 0x27b4e43c
        Reference Counts: 4:0:0, Magic: 8
3       Next Hop  :2003::7            Metric   :1               Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :700 (length 3)
        Address: 0x27c9628c  Hash:497 (0x0300067a)
        Links: 0x00000000, 0x00000000, nlri: 0x27b4e496
        Reference Counts: 1:0:0, Magic: 10
```

Syntax: show ipv6 mbgp attribute-entries

Displaying dampened paths

To display MBGP dampened paths for IPv4.

```
NetIron#show ip mbgp dampened-paths
    Status Code  >:best d:damped h:history *:valid
    Network           From            Flp Since    Reuse   Pnlty rIdx dBlk
*d  172.108.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.101.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.106.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.10.1.0/24     172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.104.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.109.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.107.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.105.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.110.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
*d  172.103.1.0/24    172.4.4.1       5   0 :2 :31 0 :41:50 3590  3   0
```

Syntax: show ip dampened-paths

To display MBGP dampened paths for IPv6.

```
NetIron#show ipv6 mbgp dampened-paths
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flaps Since    Reuse    Path
*d  2010::1/128        2004::1         3     0 :2 :25 0 :37:10 100
*d  1005::/64          2004::1         3     0 :2 :25 0 :37:10 100
*d  1003::/64          2004::1         3     0 :2 :25 0 :37:10 100
*d  1008::/64          2004::1         3     0 :2 :25 0 :37:10 100
*d  2008::/64          2004::1         3     0 :2 :25 0 :37:10 100
*d  1001::/64          2004::1         3     0 :2 :25 0 :37:10 100
*d  1006::/64          2004::1         3     0 :2 :25 0 :37:10 100
*d  1010::/64          2004::1         3     0 :2 :25 0 :37:10 100
*d  1004::/64          2004::1         3     0 :2 :25 0 :37:10 100
*d  2004::/64          2004::1         3     0 :2 :25 0 :37:10 100
```

Syntax: show ipv6 mbgp dampened-paths

Displaying MBGP filtered routes

To display MBGP filtered routes for IPv4.

```
NetIron#show ip mbgp filtered-routes
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m: NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      7.7.7.7/32         172.4.3.7       0          100        0      EF
         AS_PATH: 700
2      137.168.1.0/24     172.4.3.7       1          100        0      EF
         AS_PATH: 700
3      172.4.3.0/24       172.4.3.7       0          100        0      EF
         AS_PATH: 700
4      192.4.2.0/24       172.4.3.7       0          100        0      EF
         AS_PATH: 700
```

Syntax: show ip mbgp filtered-routes

To display MBGP filtered routes for IPv6.

```
NetIron#show ipv6 mbgp filtered-routes
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m: NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      2003::/64          2003::7         0          100        0      EF
         AS_PATH: 700
2      2017::1/128        2003::7         0          100        0      EF
         AS_PATH: 700
3      2020::/64          2003::7         0          100        0      EF
         AS_PATH: 700
4      2070::/64          2003::7         0          100        0      EF
         AS_PATH: 700
```

Syntax: show ipv6 mbgp filtered-routes

Displaying MBGP flap statistics

To display MBGP flap statistics for IPv4.

```
NetIron#show ip mbgp flap-statistics
Total number of flapping routes: 10
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flp Since    Reuse   Pnlty rIdx dBlk
*d  172.108.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.101.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.106.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.10.1.0/24      172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.104.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.109.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.107.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.105.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.110.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
*d  172.103.1.0/24     172.4.4.1       5   0 :2 :15 0 :42:0  3621  3   0
```

Syntax: show ip mbgp flap-statistics

To display MBGP flap statistics for IPv6.

```
NetIron#show ipv6 mbgp flap-statistics
Total number of flapping routes: 14
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flaps Since    Reuse    Path
h   2010::1/128        2004::1         1     0 :0 :49 0 :0 :0  100
h   1005::/64          2004::1         1     0 :0 :49 0 :0 :0  100
h   1003::/64          2004::1         1     0 :0 :49 0 :0 :0  100
h   1008::/64          2004::1         1     0 :0 :49 0 :0 :0  100
h   2008::/64          2004::1         1     0 :0 :49 0 :0 :0  100
h   1001::/64          2004::1         1     0 :0 :49 0 :0 :0  100
h   1006::/64          2004::1         1     0 :0 :49 0 :0 :0  100
h   1010::/64          2004::1         1     0 :0 :49 0 :0 :0  100
h   1004::/64          2004::1         1     0 :0 :49 0 :0 :0  100
h   2004::/64          2004::1         1     0 :0 :49 0 :0 :0  100
```

Syntax: show ipv6 mgbp flap-statistics

Displaying MBGP peer groups

To display MBGP Peer Groups for IPv4.

```
NetIron#show ip mbgp peer-group
1   BGP peer-group is group_one
      Address family : IPV4 Unicast
      Address family : IPV4 Multicast
      Address family : IPV6 Unicast
      Address family : IPV6 Multicast
    Route Filter Policies:
       Route-map: (out) wtest
    Members:
       IP Address: 2003::7, AS: 700
       IP Address: 2004::1, AS: 100
2   BGP peer-group is v4_group_one
      Address family : IPV4 Unicast
      Address family : IPV4 Multicast
      Address family : IPV6 Unicast
      Address family : IPV6 Multicast
    Members:
       IP Address: 172.4.3.7, AS: 700
       IP Address: 172.4.4.1, AS: 100
```

Syntax: show ip mbgp peer-group

To display the MBGP Peer Groups for IPv6.

```
NetIron#show ipv6 mbgp peer-group
1   BGP peer-group is group_one
      Address family : IPV4 Unicast
      Address family : IPV4 Multicast
      Address family : IPV6 Unicast
      Address family : IPV6 Multicast
    Route Filter Policies:
       Route-map: (out) wtest
    Members:
       IP Address: 2003::7, AS: 700
       IP Address: 2004::1, AS: 100
2   BGP peer-group is v4_group_one
      Address family : IPV4 Unicast
      Address family : IPV4 Multicast
      Address family : IPV6 Unicast
      Address family : IPV6 Multicast
    Members:
       IP Address: 172.4.3.7, AS: 700
       IP Address: 172.4.4.1, AS: 100
```

Syntax: show ipv6 mbgp peer-group

Clearing MBGP information

Use the commands in this section to clear MBGP information.

Clearing route flap dampening information

To clear MBGP IPv4 route flap dampening information, enter the following command.

```
NetIron#clear ip mbgp dampening
```

Syntax: clear ip mbgp dampening

To clear MBGP IPv6 route flap dampening information, enter the following command.

```
NetIron#clear ipv6 mbgp dampening
```

Syntax: clear ipv6 mbgp dampening

Clearing route flap statistics

To clear MBGP IPv4 route flap statistics, enter the following command.

```
NetIron#clear ip mbgp flap-statistics
```

Syntax: clear ip mbgp flap-statistics

To clear MBGP IPv6 route flap statistics, enter the following command.

```
NetIron#clear ipv6 mbgp flap-statistics
```

Syntax: clear ipv6 mbgp flap-statistics

Clearing local information

To clear MBGP IPv4 local information, enter the following command.

```
NetIron#clear ip mbgp local
```

Syntax: clear ip mbgp local

To clear MBGP IPv6 local information, enter the following command.

```
NetIron#clear ipv6 mbgp local
```

Syntax: clear ipv6 mbgp local

Clearing BGP neighbor information

To clear MBGP IPv4 BGP neighbor, enter the following command.

```
NetIron#clear ip mbgp  neighbor
```

Syntax: clear ip mbgp neighbor

To clear MBGP IPv6 BGP neighbor, enter the following command.

```
NetIron#clear ipv6 mbgp neighbor
```

Syntax: clear ipv6 mbgp neighbor

Clearing BGP routes

To clear MBGP IPv4 BGP routes, enter the following command.

```
NetIron#clear ip mbgp  routes
```

Syntax: clear ip mbgp routes

To clear MBGP IPv6 BGP routes, enter the following command.

```
NetIron#clear ipv6 mbgp  routes
```

Syntax: clear ipv6 mbgp routes

Clearing traffic counters

To clear MBGP IPv4 BGP traffic counters, enter the following command.

```
NetIron#clear ip mbgp  traffic
```

Syntax: clear ip mbgp traffic

To clear MBGP IPv6 BGP traffic counters, enter the following command.

```
NetIron#clear ipv6 mbgp  traffic
```

Syntax: clear ipv6 mbgp traffic

Clearing VPN4 address family

To clear MBGP IPv4 VPNV4 address family information, enter the following command.

```
NetIron#clear ip mbgp vpnv4
```

Syntax: clear ip mbgp vpnv4

Clearing VPN Routing/Forwarding information

To clear MBPG IPv4 VPN Routing/Forwarding (VRF) instance information, enter the following command.

```
NetIron#clear ip mbgp vrf
```

Syntax: clear ip mbgp vrf

Configuring Multi-VRF 29

Overview of Multi-VRF

The following list displays the Multi-VRF features supported by PowerConnect B-MLXe.

NOTE
Packet over Sonet (POS) modules do not support Multi-VRF.

- Multi-VRF
- Multi-VRF for IPv4 Unicast - Static routing
- Multi-VRF for IPv4 Unicast - RIP
- Multi-VRF for IPv4 Unicast - OSPF
- Multi-VRF for IPv4 Unicast - BGP
- Multi-VRF for IBGP

Note: All VRFs share the same AS number.

Virtual Private Networks (VPNs) have been a key application in networking for a long time. Many possible solutions have been proposed over the last several years. Among the many requirements driving this need have been the need for secure transport of sensitive information and controlling information access to those who need it. In large enterprises, particularly those distributed across disparate locations, sensitivity to information pertinent to a department drives the requirement for an IT manager to logically demarcate information flows to be within that department. The need for privacy is another driver behind deployment of VPN solutions.

VPN technologies can be broadly classified into 2 types:

- secure VPNs
- trusted VPNs.

Secure VPNs require traffic to be encrypted and authenticated and are most important when communication occurs across an infrastructure that is not trusted (e.g. over the public Internet). The most commonly deployed types of secure VPNs are IPSec VPNs and SSL (Secure Sockets Layer) VPNs. Both offer encryption of data streams. While IPSec VPNs operate at the network layer and require special client software, SSL VPNs are more application centric and can generally work with any SSL-enabled browser.

Trusted VPNs ensure integrity and privacy of the data transfers but do not provide any encryption capabilities. Trusted VPNs are most useful when the goal is to leverage a shared infrastructure to allow virtual networks to be built. Examples of such "trusted VPN" technologies include IP or MPLS based Layer 2 VPNs (VPLS, VLL), BGP or MPLS VPNs, ATM or Frame Relay circuits, Layer 2 Tunneling Protocol (L2TP), etc. In short, all these technologies allow a shared infrastructure to be used without compromising the privacy needs of different users or user groups.

Central to Multi-VRF is the ability to maintain multiple "Virtual Routing and Forwarding" (VRF) tables on the same Provider Edge (PE) Router. Multi-VRF uses multiple instances of a routing protocol such as BGP or OSPF to exchange route information for a VPN among peer PE routers. The Multi-VRF capable PE router maps an input customer interface to a unique VPN instance. The router maintains a different VRF table for each VPN instance on that PE router. Multiple input interfaces may also be associated with the same VRF on the router, if they connect to sites belonging to the same VPN. This input interface can be a physical interface or a virtual Ethernet interface on a port.

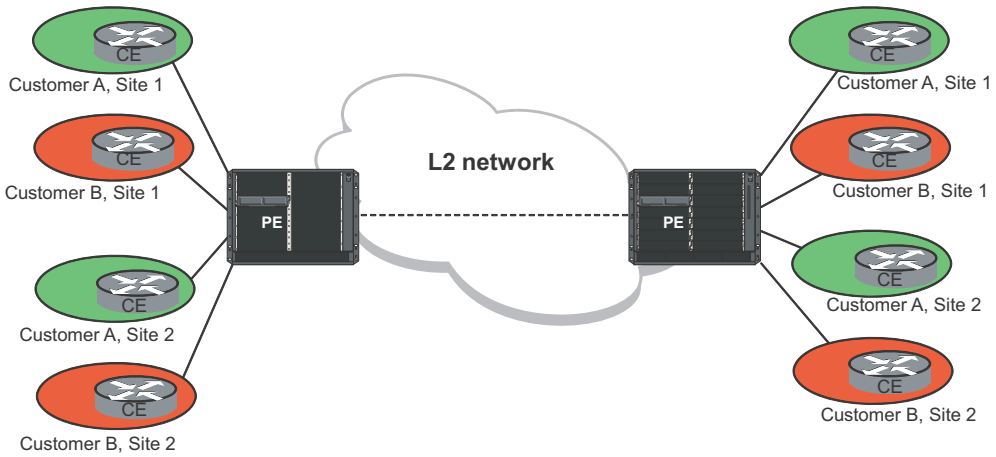
Multi-VRF routers communicate with one another by exchanging route information in the VRF table with the neighboring PE router. This exchange of information among the PE routers is done using BGP or OSPF. The PE routers that communicate with one another should be directly connected at Layer 3. Customers connect to PE routers in the network using Customer Edge (CE) routers as shown in Figure 168.

Different routing protocols may be used for exchanging information between the PE-PE routers and between the adjacent PE-CE routers. Further, different PE-CE routing protocols may be used in a VPN to exchange customer routes with the various customer sites in that VPN. The routes learned from the PE-CE protocol are added to the corresponding VRF instance and redistributed through the PE-PE protocol to the peer router in the backbone network.

Figure 168 depicts a network using Multi-VRF to provide connectivity among sites that belong to multiple VPNs. To share the VPN route table information with remote PEs, each PE creates separate virtual interfaces and run different instances of the PE-PE routing protocol for each VRF.

NOTE
Some vendors also use the terminology of "Multi-VRF CE" or "VRF-Lite" for this technology.

FIGURE 168 A Network deploying Multi-VRF



Multi-VRF and BGP or MPLS VPNs share some common aspects. For instance, in both cases the edge router maintains a VRF for all directly connected sites that are part of the same VPN. Also in both cases, the PE and CE routers share customer route information using a variety of PE-CE routing protocols, such as OSPF, RIP, E-BGP or static routes. Overlapping address spaces among different VPNs are allowed for both.

There are however, several differences between the two VPN technologies. The fundamental difference between the two technologies is that Multi-VRF requires that peering PE routers be directly connected at Layer 3. A Layer 2 network however, can be present between these directly-connected PE routers. BGP or MPLS VPNs do not have this restriction. In BGP or MPLS VPNs, the MPLS network determines the path to the peer router. In order to distinguish between devices with overlapping IP addresses, route targets are used in BGP or MPLS VPNs. Multi-VRF uses the input interface to uniquely identify the associated VPN, which is why the two PE routers should be directly connected at Layer 3. Table compares Multi-VRF and BGP or MPLS VPNs in more detail

TABLE 197 Comparison between Multi-VRF and BGP or MPLS VPNs

	Multi-VRF	**BGP or MPLS VPN**
PE-PE Routing Protocol	BGP, OSPF, RIP or Static routing	BGP
PE-CE Routing Protocol	BGP, OSPF, RIP or Static routing	BGP, OSPF, RIP or Static routing
PE-PE Routing Connectivity	PE Routers should be directly connected at Layer 3	PE Routers are interconnected through an IP or MPLS Network
Determination of VRF Instance	Based on input interface only	Based on route target (network interface) or input interface (CE)
Number of Routing Protocol Instances (PE to PE)	Unique routing protocol instance for each VRF instance	Single routing protocol instance
Controlling Advertisement of Routes	No need for route targets to be used. Advertisement on one VRF is indendant of advertisement in other VRFs.	Route targets used to identify the customer VPN in advertised routes. The destination PE filters the routes advertised from a peer PE by comparing the route target with the VPNs maintained locally on that PE.
Number of VRF Instances	Unique VRF instance for each VPN	Unique VRF instance for each VPN
Overlapping Private Addresses allowed over VPNs?	Yes	Yes
Scalability	Reasonably Scalable	Highly Scalable
MPLS Required	No	Yes

Benefits and applications of Multi-VRF

Multi-VRF provides a reliable mechanism for a network administrator to maintain multiple virtual routers on the same device. The goal of providing isolation among different VPN instances is accomplished without the overhead of heavyweight protocols used in secure VPN technologies or the administrative complexity of MPLS VPNs. It is particularly effective when operational staff has expertise in managing IP networks but may not have the same familiarity in managing MPLS networks. Overlapping address spaces can be maintained among the different VPN instances.

As the two examples in the following sections demonstrate, the simplicity of Multi-VRF allows for several interesting applications.

Example of Multi-VRF usage in an enterprise data center

Figure 169 displays an example of Multi-VRF in an enterprise data center. Each server farm is used for a dedicated application or set of applications. For security reasons, only specific servers in this farm may be allowed to communicate with other servers. Access in some cases may be completely prohibited whereas in other cases access may be allowed through the firewall. Each server is placed on a different subnet. To ensure optimal performance of the data center, trusted servers should be allowed to communicate directly whereas un-trusted servers should not be allowed to directly communicate at all. While Figure 169 shows a limited number of servers; in practice, the number of servers used for this application can run from the tens to the hundreds.

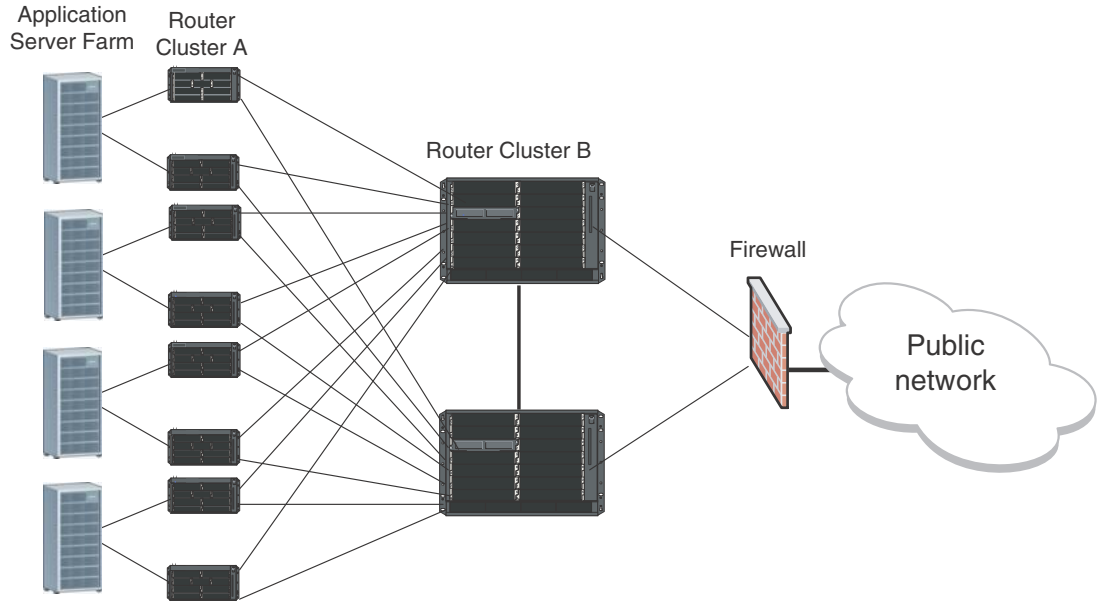
A common way to configure this example is by using Policy Based Routing (PBR). However, because PBR can become very difficult to administer and manage as the network begins to grow, it may require frequent configuration changes which is prone to introducing operator errors.

MPLS VPNs can also be used to configure this example. However, it may be too heavy-weight for what needs to be accomplished in this scenario. In addition, operational staff in enterprise data centers may not always be conversant with administering MPLS.

Secure VPN technologies like IP-Sec are not required here because the infrastructure is already secure. Therefore, the overhead of encryption is not needed.

Multi-VRF is an ideal solution for an application like this example. The servers that are allowed to communicate can be placed in the same Multi-VRF instance. If server access is to be controlled at a more granular level (e.g. at the application layer), then traffic from specific applications on that server can be sent on a specific tagged interface to the router in Cluster A. As shown in Figure 169, a highly redundant cluster is achieved by ensuring that no single node becomes a point of failure within this network.

FIGURE 169 Example of Multi-VRF usage in an enterprise data center application



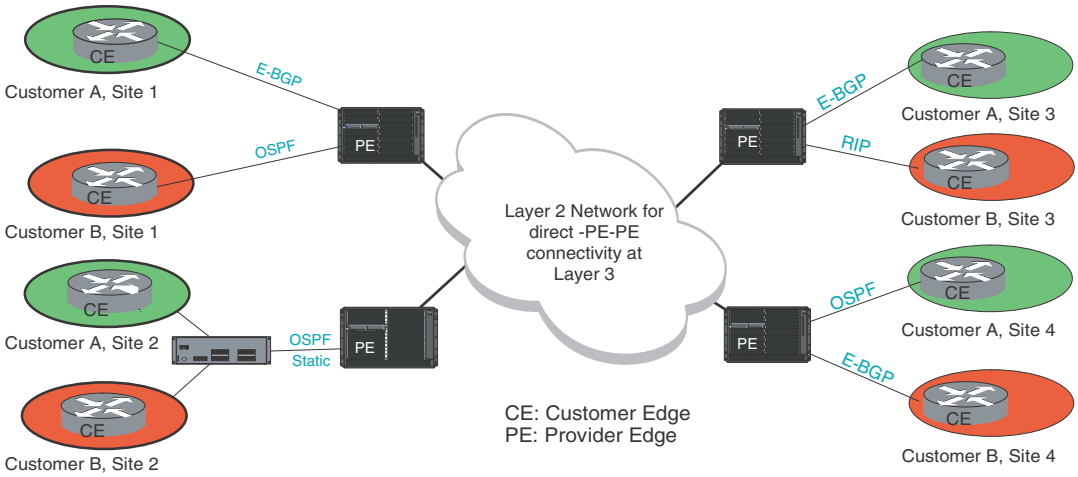
Example of Multi-VRF usage in a service provider network

Figure 170 depicts the use of Multi-VRF in a typical service provider application. This service provider owns a Layer 2 network connecting the PEs and offers managed VPN services to end users. As shown in Figure 169, a host of PE-CE routing protocols can be used-E-BGP, OSPF, RIP or Static Routing.

It is also possible that a site (such as site 2) may have several customers in close geographical proximity as in a business park. This may warrant a dedicated MTU to be placed on-site, which is owned by the service provider. In such a scenario, the different customers may share the same MTU and still use overlapping private address spaces. The MTU is a switch that adds a unique VLAN tag for each connected customer. The PE router (labeled PE2) maps a Layer 3 tagged interface to a unique VRF. Thus, it could be sharing routes using OSPF with one CE and just using Static Routing with another CE (both of these may occur over different virtual interfaces on the same physical interface).

Layer 3 BGP or MPLS VPNs could also be used in a network such as the above. However, if one of the PE routers does not support MPLS or if the operational staff is not conversant with MPLS operations, Multi-VRF provides an alternative mechanism to achieve the same objective.

FIGURE 170 Multi-VRF in a service provider application



Summary

Multi-VRF provides a reliable mechanism for trusted virtual private networks to be built over a shared infrastructure. The ability to maintain multiple virtual routing or forwarding tables allows overlapping private IP addresses to be maintained across VPNs and accomplish goals very similar to that those of more complex VPN technologies such as BGP or MPLS VPNs.

Configuring Multi-VRF

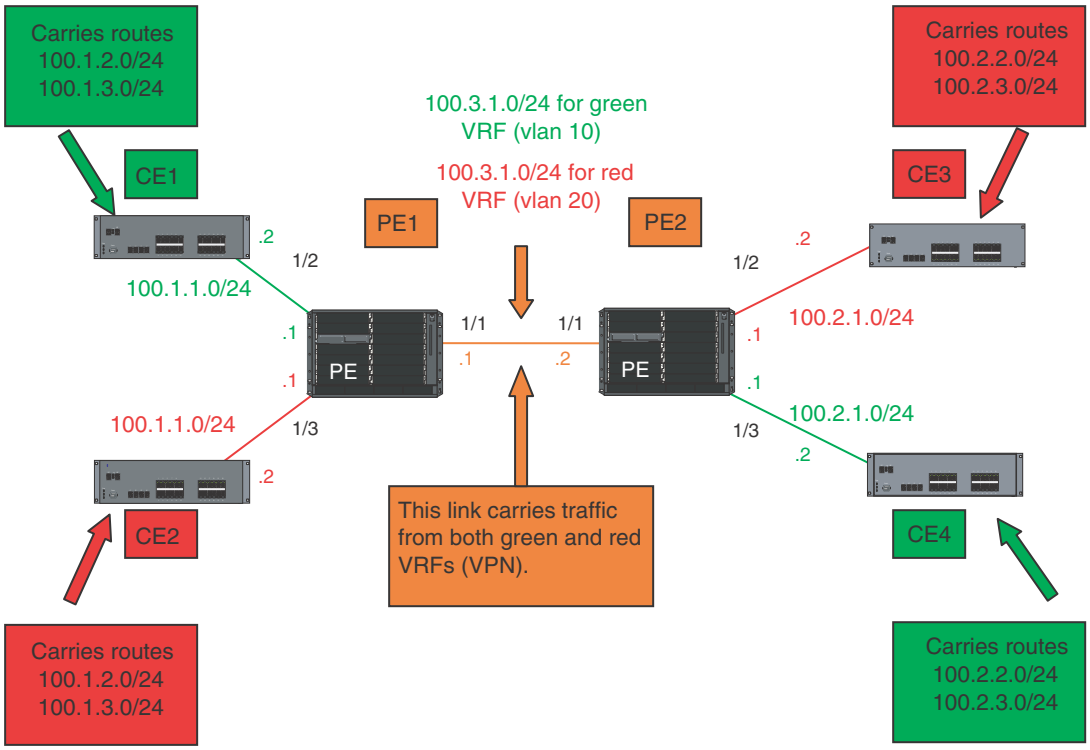
Configuration of the Multi-VRF feature uses the following commands that are defined in "Configuring BGP VPNs on a PE" on page 1603:

- "Defining a VRF routing instance" – This procedure describes how to define a VRF using the **ip vrf** command.
- "Assigning a Route Distinguisher to a VRF" – This procedure describes how to define a Route Distinguisher (RD). The RD sets a unique identity to an instance of a VRF. As such, it allows the same IP address to be used in different VPNs without creating any conflict.
- "Assigning a VRF routing instance to an interface" – This procedure describes how to assign a VRF to one or more virtual or physical interfaces.
- "Assigning a VRF routing instance to a LAG interface"– This procedure describes how to assign a VRF to a LAG interface.

The main difference between configurations described in "Configuring BGP VPNs on a PE"and Multi-VRF is that there is no MPLS configuration required for Multi-VRF. This section provides a common Multi-VRF configuration with two possible methods to achieve that configuration.

The diagram in Figure 171 shows a typical network utilizing the multi-VRF feature to implement layer 3 VPNs across 2 directly connected (at layer3) PE routers.

FIGURE 171 Example network topology with both RED and GREEN VPNs



In the diagram in Figure 171, CE1 and CE4 are customer edge (CE) routers for the “green” VPN, while CE2 and CE3 belong to “red” VPN. These CE routers can be any routers or layer 3 switches that are capable of running one or many dynamic routing protocols such as BGP, OSPF or RIP or even simple static routing.

The 2 PEs routers have to be routers that are capable of supporting VRF routing (either with or without MPLS support). They connect all four CE routers together with a single link between the two of them. Note that this single link between the 2 PEs could also be replaced by a layer-2 switched network if direct physical connection between the PEs is not possible. The only requirement for the connections is that the 2 PEs have to be “directly connected” at layer 3.

Both the customer RED and customer GREEN networks (or VPN) consist of internal routes with overlapping IP address ranges. Thus, traffic communication within each customers’ VPN across the 2 PE routers, i.e. between CE1 and CE4, and between CE2 and CE3, must be separated using VRFs.

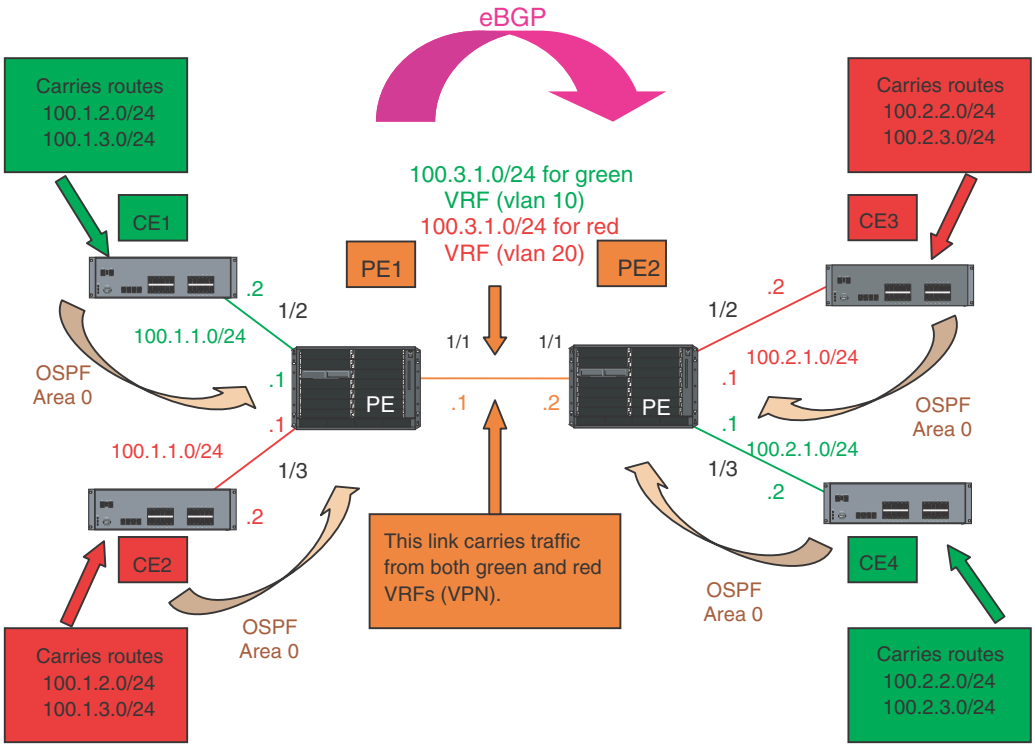
The following sections provide two examples of how to set-up the network shown in Figure 171 using different routing protocol configurations:

- **Configuration 1** – eBGP Configured between PE1 and PE2 with OSPF (Area 0) Configured between PEs and CEs
- **Configuration 2** – OSPF (Area 0) Configured between PE1 and PE2 with OSPF (Area 1 and Area 2) Configured between PEs and CEs

Configuration 1

As shown in Figure 172, eBGP is configured between PE1 and PE2 and OSPF (Area 0) is configured between PEs and CEs.

FIGURE 172 eBGP configured between PE1 and PE2 with OSPF (Area 0) configured between PEs and CEs



The following configuration examples for PE1, PE2, CE1, CE2, CE3, and CE4 describe how to create the example shown in Figure 172.

PE1 configuration

In this configuration, VLANs 10 and 20 are created as a link on a tagged port (e 1/10) between PE1 and PE2. Two VRFs (“RED” and “GREEN”) are then defined with each having a unique Route Distinguisher (RD). VRF “Green” is assigned an RD value of 10:10, and VRF “Red” is assigned an RD value of 20:20.

In the BGP configuration, PE1 is defined in Local AS1. VRFs “Green” and “Red” are configured and both “Green” and “Red” have the same IP network address assigned (100.3.1.2/24). This is possible because each of the BGP VRF instances have their own seperate BGP tables. This is also the same IP network address that will be assigned to VRFs “Green” and “Red” on PE2 within Local AS 2. Redistribution of OSPF routes from PE1’s CE peers is enabled to all for their advertisement to PE2.

Both VRFs are configured in Area “0” and directed to redistribute their routes to BGP. The physical interfaces (e 1/2 and e 1/3) to the CEs are assigned to the correct VRF and are configured with the same IP address (100.1.1.1/24) and OSPF Area “0”.

The virtual Interfaces (ve10 and ve20) are configured with the same IP address (100.3.1.1/24) and for IP VRF forwarding in the appropriate VRF (Green or Red).

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# tagged e 1/1
NetIron(config-vlan-10)# router-interface ve 10
NetIron(config-vlan-10)# vlan 20
NetIron(config-vlan-20)# tagged e 1/1
NetIron(config-vlan-20)# router-interface ve 20
NetIron(config-vlan-20)# exit
NetIron(config)# ip vrf green
NetIron(config-ip-vrf-green) rd 10:10
NetIron(config-ip-vrf-green) ip vrf red
NetIron(config-ip-vrf-red) rd 20:20
NetIron(config-ip-vrf-red) exit-vrf
NetIron(config)# router bgp
NetIron(config-bgp)# local-as 1
NetIron(config-bgp)# address-family ipv4 unicast vrf green
NetIron(config-bgp-ipv4u-vrf)# neighbor 100.3.1.2 remote-as 2
NetIron(config-bgp-ipv4u-vrf)# network 100.3.1.0/24
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match internal
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match external1
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match external2
NetIron(config-bgp-ipv4u-vrf)# exit
NetIron(config)# router bgp
NetIron(config-bgp)# address-family ipv4 unicast vrf red
NetIron(config-bgp-ipv4u-vrf)# neighbor 100.3.1.2 remote-as 2
NetIron(config-bgp-ipv4u-vrf)# network 100.3.1.0/24
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match internal
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match external1
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match external2
NetIron(config-bgp-ipv4u-vrf)# exit
NetIron(config)# router ospf vrf green
NetIron(config-ospf-router-vrf-green)# area 0
NetIron(config-ospf-router-vrf-green)# redistrbution bgp
NetIron(config-ospf-router-vrf-green)# exit
NetIron(config)# router ospf vrf red
NetIron(config-ospf-router-vrf-red)# area 0
NetIron(config-ospf-router-vrf-red)# redistrbution bgp
NetIron(config-ospf-router-vrf-red)# exit
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e1000-1/2)# ip vrf forwarding green
NetIron(config-if-e1000-1/2)# ip ospf area 0
NetIron(config-if-e1000-1/2)# ip address 100.1.1.1/24
NetIron(config-if-e1000-1/2)# interface ethernet 1/3
NetIron(config-if-e1000-1/3)# ip vrf forwarding red
NetIron(config-if-e1000-1/3)# ip ospf area 0
NetIron(config-if-e1000-1/3)# ip address 100.1.1.1/24
NetIron(config-if-e1000-1/3)# exit
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ip vrf forwarding green
NetIron(config-vif-10)# ip address 100.3.1.1/24
NetIron(config-vif-10)# Interface ve 20
NetIron(config-vif-10)# ip vrf forwarding red
NetIron(config-vif-10)# ip address 100.3.1.1/24
```

PE2 Configuration:

The PE2 configuration is a mirror image of the PE1 configuration. The only difference is that the BGP neighbor is port 1/1 on PE1 which has an IP address of 100.3.1.1. This is used in the BGP configuration.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# tagged e 1/1
NetIron(config-vlan-10)# router-interface ve 10
NetIron(config-vlan-10)# vlan 20
NetIron(config-vlan-20)# tagged e 1/1
NetIron(config-vlan-20)# router-interface ve 20
NetIron(config-vlan-20)# exit-vrf
NetIron(config)# ip vrf green
NetIron(config-ip-vrf-green) rd 10:10
NetIron(config-ip-vrf-green) ip vrf red
NetIron(config-ip-vrf-red) rd 20:20
NetIron(config-ip-vrf-red) exit
NetIron(config)# router bgp
NetIron(config-bgp)# local-as 1
NetIron(config-bgp)# address-family ipv4 unicast vrf green
NetIron(config-bgp-ipv4u-vrf)# neighbor 100.3.1.1 remote-as 2
NetIron(config-bgp-ipv4u-vrf)# network 100.3.1.0/24
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match internal
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match external1
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match external2
NetIron(config-bgp-ipv4u-vrf)# exit
NetIron(config)# router bgp
NetIron(config-bgp)# address-family ipv4 unicast vrf red
NetIron(config-bgp-ipv4u-vrf)# neighbor 100.3.1.1 remote-as 2
NetIron(config-bgp-ipv4u-vrf)# network 100.3.1.0/24
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match internal
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match external1
NetIron(config-bgp-ipv4u-vrf)# redistribute ospf match external2
NetIron(config-bgp-ipv4u-vrf)# exit
NetIron(config)# router ospf vrf green
NetIron(config-ospf-router-vrf-green)# area 0
NetIron(config-ospf-router-vrf-green)# redistrbution bgp
NetIron(config-ospf-router-vrf-green)# exit
NetIron(config)# router ospf vrf red
NetIron(config-ospf-router-vrf-red)# area 0
NetIron(config-ospf-router-vrf-red)# redistrbution bgp
NetIron(config-ospf-router-vrf-red)# exit
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e1000-1/2)# ip vrf forwarding green
NetIron(config-if-e1000-1/2)# ip ospf area 0
NetIron(config-if-e1000-1/2)# ip address 100.1.1.1/24
NetIron(config-if-e1000-1/2)# interface ethernet 1/3
NetIron(config-if-e1000-1/3)# ip vrf forwarding red
NetIron(config-if-e1000-1/3)# ip ospf area 0
NetIron(config-if-e1000-1/3)# ip address 100.1.1.1/24
NetIron(config-if-e1000-1/3)# exit
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ip vrf forwarding green
NetIron(config-vif-10)# ip address 100.3.1.1/24
NetIron(config-vif-10)# Interface ve 20
NetIron(config-vif-10)# ip vrf forwarding red
NetIron(config-vif-10)# ip address 100.3.1.1/24
```

CE 1 and CE 2 configurations

The CE1 and CE2 router configurations are exactly the same. Both are configured in OSPF Area 0 with route redistribution enabled. The IP addresses: 100.1.2.1/32 and 100.1.3.1/32 are configured for the Loopback1 interface allowing them to carry routes from these networks.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 0
NetIron(config-ospf-router)# redistribution connected
NetIron(config-ospf-router)# exit
NetIron(config)# interface loopback 1
NetIron(config-lbif-1)# ip address 100.1.2.1/32
NetIron(config-lbif-1)# ip address 100.1.3.1/32
NetIron(config-lbif-1)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# ip ospf area 0
NetIron(config-if-e1000-1/1)# ip address 100.1.1.2/24
```

CE 3 and CE 4 configurations

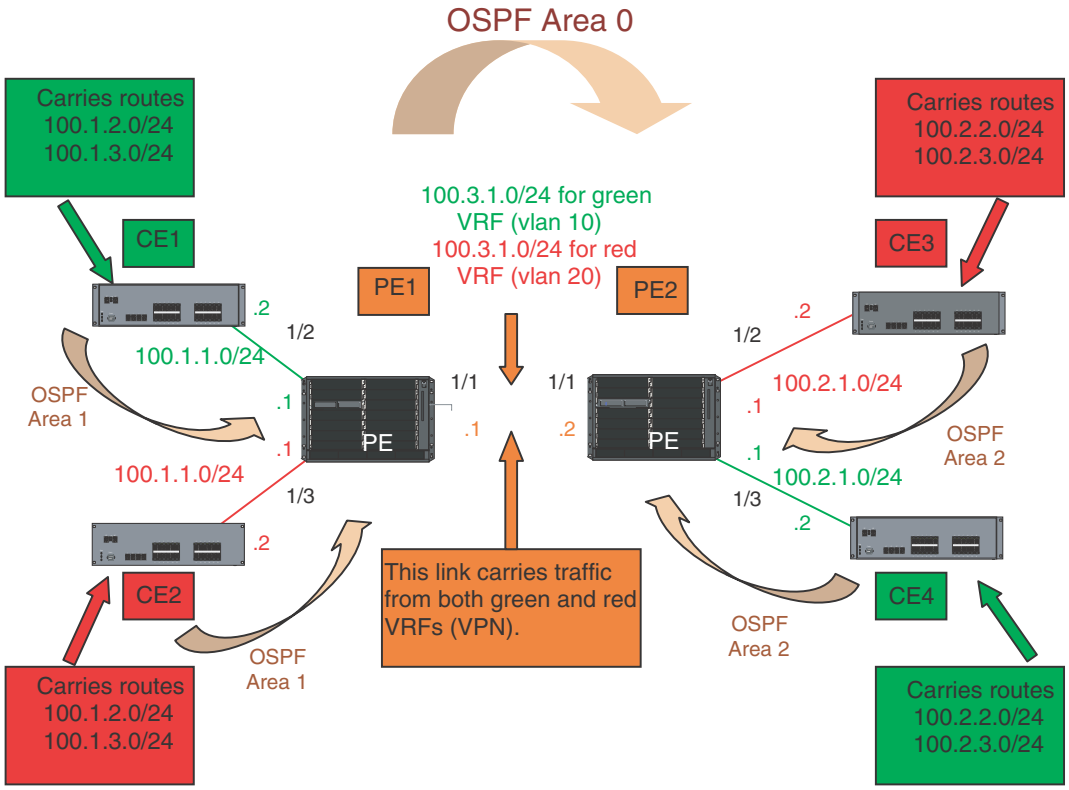
The CE3 and CE4 router configurations are exactly the same. Both are configured in OSPF Area 0 with route redistribution enabled. The IP addresses: 100.2.2.1/32 and 100.2.3.1/32 are configured for the Loopback1 interface allowing them to carry routes from these networks.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 0
NetIron(config-ospf-router)# redistribution connected
NetIron(config-ospf-router)# exit
NetIron(config)# interface loopback 1
NetIron(config-lbif-1)# ip address 100.2.2.1/32
NetIron(config-lbif-1)# ip address 100.2.3.1/32
NetIron(config-lbif-1)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# ip ospf area 0
NetIron(config-if-e1000-1/1)# ip address 100.2.1.2/24
```

Configuration 2

As shown in Figure 172, OSPF (Area 0) is configured between PE1 and PE2 and OSPF (Area 1 and Area 2) is configured between PEs and CEs. The biggest difference between this configuration and Configuration 1 is that OSPF is used between the 2 PEs instead of eBGP. Otherwise most of the configuration is the same as Configuration 1.

FIGURE 173 OSPF (Area 0) configured between PE1 and PE2 with OSPF (Area 1 and Area 2) configured between PEs and CEs



The following configuration examples for PE1, PE2, CE1, CE2, CE3, and CE4 describe how to create the example shown in Figure 173.

PE1 configuration:

In this configuration, VLANs 10 and 20 are created as a link on a tagged port (e 1/10) between PE1 and PE2. Two VRFs (“RED” and “GREEN”) are then defined with each having a unique Route Distinguisher (RD). VRF “Green” is assigned an RD value of 10:10, and VRF “Red” is assigned an RD value of 20:20.

Because OSPF is the only routing protocol used in this set-up, multiple OSPF areas are used. Area 0 is configured between the two PEs. Area 1 is configured PE1 and CE’s 1 and 2. Area 0 is configured between the two PEs. Area 2 is configured PE2 and CE’s 3 and 4.

The virtual Interfaces (ve10 and ve20) are configured with the same IP address (100.3.1.1/24) and for IP VRF forwarding in the appropriate VRF (Green or Red). Both are also configured in OSPF Area 0.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# tagged e 1/1
NetIron(config-vlan-10)# router-interface ve 10
NetIron(config-vlan-10)# vlan 20
NetIron(config-vlan-20)# tagged e 1/1
NetIron(config-vlan-20)# router-interface ve 20
NetIron(config-vlan-20)# exit-vrf
NetIron(config)# ip vrf green
```

```
NetIron(config-ip-vrf-green) rd 10:10
NetIron(config-ip-vrf-green) ip vrf red
NetIron(config-ip-vrf-red) rd 20:20
NetIron(config-ip-vrf-red) exit-vrf
NetIron(config)# router ospf vrf green
NetIron(config-ospf-router-vrf-green)# area 0
NetIron(config-ospf-router-vrf-green)# area 1
NetIron(config-ospf-router-vrf-green)# exit
NetIron(config)# router ospf vrf red
NetIron(config-ospf-router-vrf-red)# area 0
NetIron(config-ospf-router-vrf-red)# area 1
NetIron(config-ospf-router-vrf-red)# exit
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e1000-1/2)# ip vrf forwarding green
NetIron(config-if-e1000-1/2)# ip ospf area 1
NetIron(config-if-e1000-1/2)# ip address 100.1.1.1/24
NetIron(config-if-e1000-1/2)# interface ethernet 1/3
NetIron(config-if-e1000-1/3)# ip vrf forwarding red
NetIron(config-if-e1000-1/3)# ip ospf area 1
NetIron(config-if-e1000-1/3)# ip address 100.1.1.1/24
NetIron(config-if-e1000-1/3)# exit
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ip vrf forwarding green
NetIron(config-vif-10)# ip ospf area 0
NetIron(config-vif-10)# ip address 100.3.1.1/24
NetIron(config-vif-10)# Interface ve 20
NetIron(config-vif-20)# ip vrf forwarding red
NetIron(config-vif-20)# ip ospf area 0
NetIron(config-vif-20)# ip address 100.3.1.1/24
```

PE2 configuration:

The PE2 configuration is a mirror image of the PE1 configuration. The only difference is that PE2 connects to CE3 and CE 4 in OSPF Area 2

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# tagged e 1/1
NetIron(config-vlan-10)# router-interface ve 10
NetIron(config-vlan-10)# vlan 20
NetIron(config-vlan-20)# vlan 20
NetIron(config-vlan-20)# tagged e 1/1
NetIron(config-vlan-20)# router-interface ve 20
NetIron(config-vlan-20)# exit
NetIron(config)# ip vrf green
NetIron(config-ip-vrf-green) rd 10:10
NetIron(config-ip-vrf-green) ip vrf red
NetIron(config-ip-vrf-red) rd 20:20
NetIron(config-ip-vrf-red) exit
NetIron(config)# router ospf vrf green
NetIron(config-ospf-router-vrf-green)# area 0
NetIron(config-ospf-router-vrf-green)# area 2
NetIron(config-ospf-router-vrf-green)# exit
NetIron(config)# router ospf vrf red
NetIron(config-ospf-router-vrf-red)# area 0
NetIron(config-ospf-router-vrf-red)# area 2
NetIron(config-ospf-router-vrf-red)# exit
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e1000-1/2)# ip vrf forwarding red
NetIron(config-if-e1000-1/2)# ip ospf area 2
```

```
NetIron(config-if-e1000-1/2)# ip address 100.2.1.1/24
NetIron(config-if-e1000-1/2)# interface ethernet 1/3
NetIron(config-if-e1000-1/3)# ip vrf forwarding green
NetIron(config-if-e1000-1/3)# ip ospf area 2
NetIron(config-if-e1000-1/3)# ip address 100.2.1.1/24
NetIron(config-if-e1000-1/3)# exit
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ip vrf forwarding green
NetIron(config-vif-10)# ip ospf area 0
NetIron(config-vif-10)# ip address 100.3.1.1/24
NetIron(config-vif-10)# Interface ve 20
NetIron(config-vif-20)# ip vrf forwarding red
NetIron(config-vif-20)# ip ospf area 0
NetIron(config-vif-20)# ip address 100.3.1.1/24
```

CE 1 and CE 2 configurations

The CE1 and CE2 router configurations are exactly the same. Both are configured in OSPF Area 1 with route redistribution enabled. The IP addresses: 100.1.2.1/24 and 100.1.3.1/24 are configured for the Loopback1 interface allowing them to carry routes from these networks.

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 1
NetIron(config-ospf-router)# redistribution connected
NetIron(config-ospf-router)# exit
NetIron(config)# interface loopback 1
NetIron(config-lbif-1)# ip address 100.1.2.1/24
NetIron(config-lbif-1)# ip address 100.1.3.1/24
NetIron(config-lbif-1)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# ip ospf area 1
NetIron(config-if-e1000-1/1)# ip address 100.1.1.2/24
```

CE 3 and CE 4 configurations

The CE3 and CE4 router configurations are exactly the same. Both are configured in OSPF Area 2 with route redistribution enabled. The IP addresses: 100.2.2.1/24 and 100.2.3.1/24 are configured for the Loopback1 interface allowing them to carry routes from these networks

```
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 2
NetIron(config-ospf-router)# redistribution connected
NetIron(config-ospf-router)# exit
NetIron(config)# interface loopback 1
NetIron(config-lbif-1)# ip address 100.1.2.1/24
NetIron(config-lbif-1)# ip address 100.1.3.1/24
NetIron(config-lbif-1)# interface ethernet 1/1
NetIron(config-if-e1000-1/1)# ip ospf area 2
NetIron(config-if-e1000-1/1)# ip address
```

Configuring MPLS Traffic Engineering 30

Overview

The following list displays the MPLS Traffic Engineering features supported by PowerConnect B-MLXe:

- Multiprotocol Label Switching (MPLS)
- MPLS Traffic Engineering
- OSPF-TE Link State Advertisements for MPLS Interfaces
- MPLS Traffic Engineering - OSPF-TE
- MPLS Traffic Engineering - ISIS-TE
- ISIS Link State Protocol data units with TE Extensions for MPLS Interfaces
- RSVP Message Authentication
- MPLS over Virtual Ethernet Interfaces
- MPLS Signalling: LDP and RSVP-TE support
- New encryption code for passwords, authentication keys, and community strings
- Traffic Engineering Database
- MPLS Fast Reroute Using One-to-One Backup
- Resetting LSPs
- Adaptive LSPs: timer-triggered LSP optimization
- Hot-standby LSPs
- RSVP Message Authentication
- Signalled LSPs
- LSP Accounting
- MPLS BFD
- Traps and Syslogs for LSPs
- Show Command to Display TE path
- Enhancements to MPLS path and route display
- Display changes for MPLS show commands for long LSP and Path names

This chapter explains how to configure ***Multiprotocol Label Switching (MPLS)*** on the PowerConnect for traffic engineering purposes. MPLS can be used to direct packets through a network over a predetermined path of routers. Forwarding decisions in MPLS are based on the contents of a label applied to the packet.

Traffic engineering is the ability to direct packets through a network efficiently, using information gathered about network resources. When used as an application of MPLS, traffic engineering involves creating paths that make the best use of available network resources, avoiding points of congestion and making efficient use of high bandwidth interfaces. Packets travelling over these paths are forwarded using MPLS.

IETF RFC and Internet draft support

The implementation of MPLS supports the following IETF RFCs and Internet Drafts.

MPLS

RFC 3031 – Multiprotocol Label Switching Architecture

RFC 3032 – MPLS Label Stack Encoding

RFC 3036 – LDP Specification

RFC 2205 – Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification

RFC 2209 – Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rule

RFC 3209 – RSVP-TE

RFC 3270 – MPLS Support of Differentiated Services

RFC 4090 – Facility backup and Fast Reroute

OSPF

RFC 3630 TE Extensions to OSPF v2

ISIS

RFC 3784 Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)

How MPLS works

MPLS uses a ***label switching*** forwarding method to direct packets through a network. In label switching, a packet is assigned a label and passes along a predetermined path of routers. Forwarding decisions are based on the contents of the label, rather than information in the packet's IP header.

The following sections describe these basic MPLS concepts:

- How packets are forwarded through an MPLS domain
- The kinds of Label Switched Paths (LSPs) that can be configured on a device
- The components of an MPLS label header

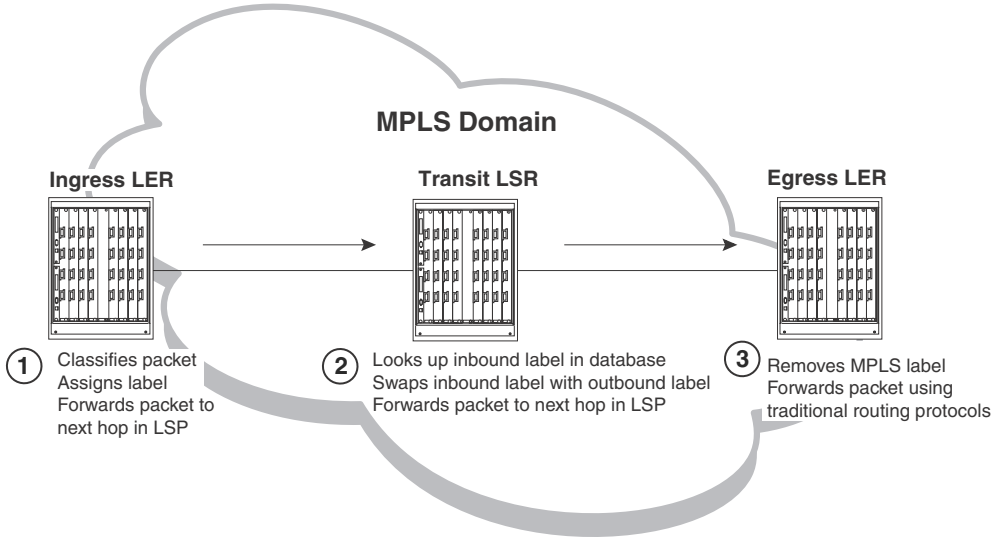
How packets are forwarded through an MPLS domain

An ***MPLS domain*** consists of a group of MPLS-enabled routers, called ***LSRs*** (Label Switching Routers). In an MPLS domain, packets are forwarded from one MPLS-enabled router to another along a predetermined path, called an ***LSP*** (Label Switched Path). LSPs are one-way paths between MPLS-enabled routers on a network. To provide two-way traffic, you configure LSPs in each direction.

The LSRs at the headend and tailend of an LSP are known as ***LERs*** (Label Edge Routers). The LER at the headend, where packets enter the LSP, is known as the ***ingress LER***. The LER at the tailend, where packets exit the LSP, is known as the ***egress LER***. Each LSP has one ingress LER and one egress LER. Packets in an LSP flow in one direction: from the ingress LER towards the egress LER. In between the ingress and egress LERs there may be zero or more ***transit LSRs***. A device enabled for MPLS can perform the role of ingress LER, transit LSR, or egress LER in an LSP. Further, a device can serve simultaneously as an ingress LER for one LSP, transit LSR for another LSP, and egress LER for some other LSP.

Figure 174 depicts an MPLS domain with a single LSP consisting of three LSRs: an ingress LER, a transit LSR, and an egress LER.

FIGURE 174 Label switching in an MPLS domain



Label switching in an MPLS domain works as described below.

1. The Ingress LER receives a packet and pushes a label onto it.

   When a packet arrives on an MPLS-enabled interface, the device determines to which LSP (if any) the packet should be assigned. Specifically, the device determines to which Forwarding Equivalence Class (FEC) the packet belongs. An FEC is simply a group of packets that should all be forwarded in the same way. For example, a FEC could be defined as all packets from a given Virtual Leased Line. FECs are mapped to LSPs. If a packet belongs to a FEC, and an LSP is mapped to that FEC, the packet is assigned to the LSP.

When a packet is assigned to an LSP, the device, acting as an ingress LER, applies (pushes) a tunnel ***label*** onto the packet. A label is a 32-bit, fixed-length identifier that is significant only to MPLS. Refer to "MPLS label header encoding" on page 1296 for specific information about the contents of a label. From this point until the packet reaches the egress LER at the end of the path, the packet is forwarded using information in its label, not information in its IP header. The packet's IP header is not examined again as long as the packet traverses the LSP. The ingress LER may also apply a VC label onto the packet based on the VPN application.

On the ingress LER, the label is associated with an outbound interface. After receiving a label, the packet is forwarded over the outbound interface to the next router in the LSP.

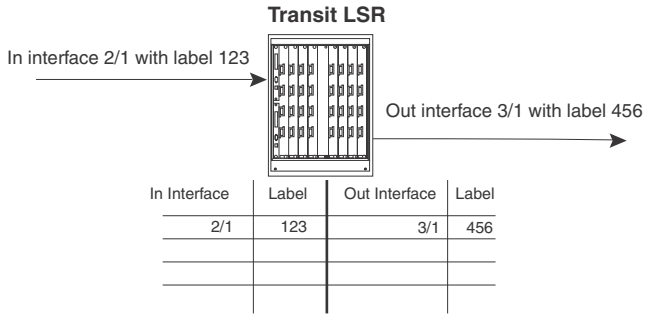
2. A transit LSR receives the labelled packet, swaps the label, and forwards the packet to the next LSR.

   In an LSP, zero or more transit LSRs can exist between the ingress and egress LERs. A transit LSR swaps labels on an MPLS packet and forwards the packet to the next router in the LSP.

   When a transit LSR receives an MPLS packet, it looks up the label in its ***MPLS forwarding table***. This table maps the label and inbound interface to a new label and outbound interface. The transit LSR replaces the old label with the new label and sends the packet out the outbound interface specified in the table. This process repeats at each transit LSR until the packet reaches the next-to-last LSR in the LSP (for signalled LSPs).

   Figure 175 shows an example of the label swapping process on a transit LSR.

FIGURE 175 Label swapping on a transit LSR



In this example, a packet comes into interface 2/1 with label 123. The transit LSR then looks up this interface-label pair in its MPLS forwarding table. The inbound interface-label pair maps to an outbound-interface-label pair – in this example, interface 3/1 with label 456. The LSR swaps label 123 with label 456 and forwards the packet out interface 3/1.

3. The egress LER receives labelled packet, pops label, and forwards IP packet.

   When the packet reaches the egress LER, the MPLS label is removed (called ***popping*** the label), and the packet can then be forwarded to its destination using standard hop-by-hop routing protocols. On signalled LSPs, the label is popped at the penultimate (next to last) LSR, rather than the egress LER. Refer to "Penultimate hop popping" on page 1295 for more information.

Types of LSPs

An LSP in an MPLS domain can be either ***static*** or ***signalled***.

Signalled LSPs

Signalled LSPs are configured at the ingress LER only. When the LSP is enabled, RSVP signalling messages travel to each LSR in the LSP, reserving resources and causing labels to be dynamically associated with interfaces. When a packet is assigned to a signalled LSP, it follows a pre-established path from the LSP's ingress LER to its egress LER. This path can be one of the following:

- A path that traverses an explicitly specified set of MPLS routers
- The IGP shortest path across the MPLS domain, determined from local routing tables
- A traffic-engineered path calculated by the device using constraints such as bandwidth reservations, administrative groups, and network topology information

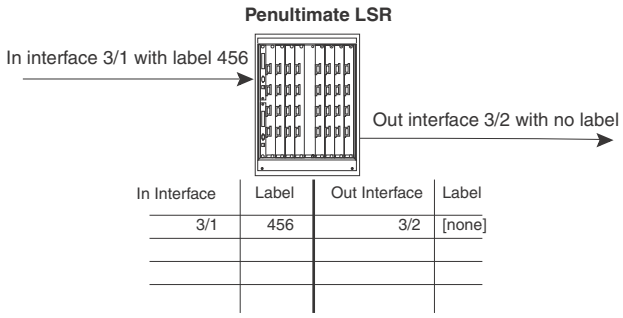
For more information, refer to "How CSPF calculates a traffic-engineered path" on page 1301, "How RSVP establishes a signalled LSP" on page 1302, and "Setting up signalled LSPs" on page 1333.

Penultimate hop popping

On signalled LSPs, the MPLS label is popped at the next-to-last LSR in the LSP, instead of at the egress LER. This action is called ***penultimate hop popping***. Penultimate hop popping improves forwarding efficiency by allowing the egress LER to avoid performing both a MPLS forwarding table lookup and an IP forwarding table lookup for each packet exiting the LSP. Instead, the MPLS label is popped at the penultimate (next-to-last) LSR, and the packet is forwarded to the egress LER with no MPLS encoding. The egress LER, in fact, does not recognize the packet as emerging from an LSP.

Figure 176 illustrates the operation that takes place at the penultimate LSR in an LSP.

FIGURE 176 Penultimate hop popping



When an LSR receives an MPLS packet, it looks up the label in its MPLS forwarding table. Normally, this table maps the label and inbound interface to a new label and outbound interface. However, when this is the penultimate LSR in an LSP, the label and inbound interface map only to an outbound interface. The penultimate LSR pops the label and forwards the packet – now a regular IP packet – out the outbound interface. When the packet reaches the egress LER, there is no indication that it had been forwarded over an LSP. The packet is forwarded using standard hop-by-hop routing protocols.

NOTE
Penultimate hop popping is always performed on signalled LSPs.

MPLS label header encoding

The following diagram illustrates the structure of the 32-bit MPLS label header. When a packet enters an LSP, the ingress LER pushes a label onto the packet.

FIGURE 177 Structure of an MPLS Label Header

An MPLS label header is composed of the following parts:

Label value (20 bits)

The label value is an integer in the range 16 – 1048575. (Labels 0 – 15 are reserved by the IETF for special usage.) For signalled LSPs, the device dynamically assigns labels in the range 1024 – 499999.

EXP field (3 bits)

The EXP field is designated for experimental usage. By default, a device uses the EXP field to define a Class of Service (CoS) value for prioritizing packets travelling through an LSP. Please refer to 9, "Configuring Quality of Service for the NetIron MLX", for more information. Please note that software forwarded VPLS packets do not use the EXP encode table.

S (Bottom of Stack) field (1 bit)

An MPLS packet can be assigned multiple labels. If an MPLS packet has multiple labels, they are logically organized in a last-in, first-out ***label stack***. An LSR performs a pop or swap operation on the topmost label; that is, the most recently applied label in the stack. The Bottom of Stack field indicates whether this label is the last (oldest) label in the stack. If the label is the last one in the stack, the Bottom of Stack field is set to 1. If not, the Bottom of Stack field is set to 0.

A device acting as an LSR can perform one push, swap, or pop operation on an incoming MPLS packet. The device can accept MPLS packets that contain multiple labels, but only the topmost label is acted upon.

TTL field (8 bits)

The TTL field indicates the Time To Live (TTL) value for the MPLS packet. At the ingress LER, an IP packet's TTL value is copied to its MPLS TTL field. At each transit LSR hop, the MPLS TTL value is decremented by 1. If the MPLS TTL value reaches 0, the packet is discarded. Optionally, you can configure the LSRs not to decrement the MPLS TTL value at each hop.

Using MPLS in traffic engineering

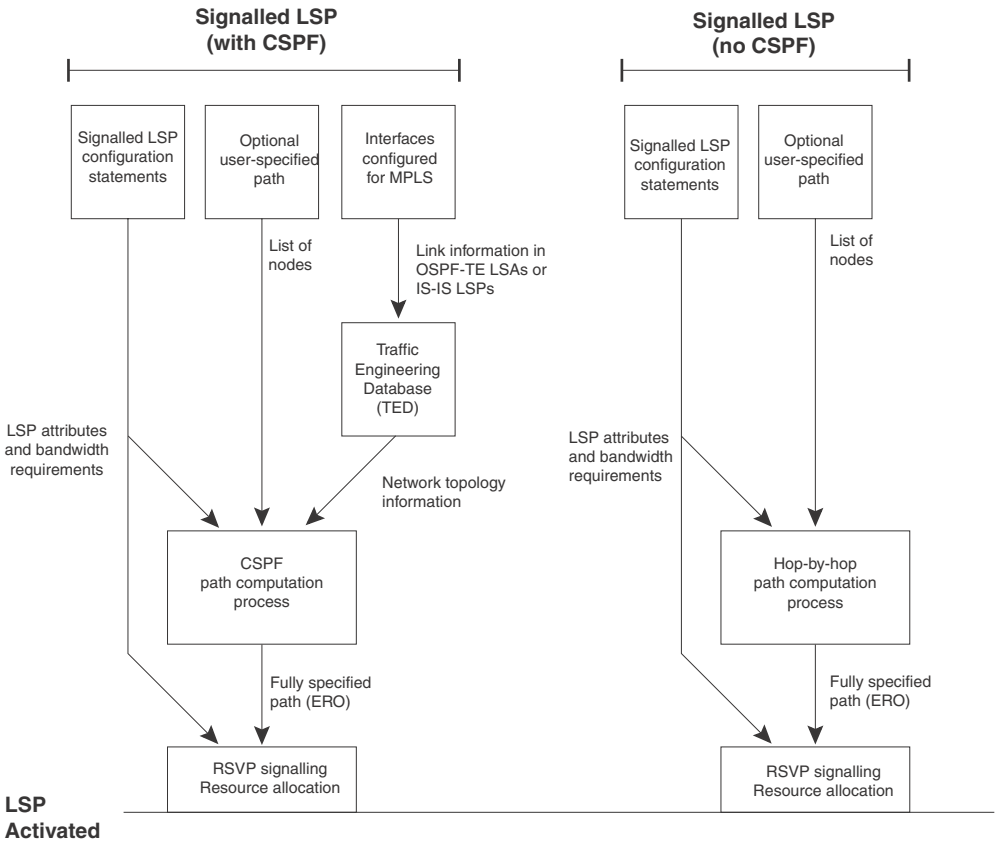
Traffic engineering is the task of routing network traffic to avoid points of congestion and make efficient use of high bandwidth interfaces. When used as an application of MPLS, traffic engineering involves creating LSPs that make the best use of available network resources; that is, ***traffic-engineered LSPs***. This section explains the process of creating traffic-engineered LSPs.

Creating traffic-engineered LSPs involves the following tasks:

- Gathering information about the network
- Using the gathered information to select optimal paths through the network
- Setting up and maintaining the paths

For traffic-engineered signalled LSPs, devices can perform these tasks dynamically. Figure 178 illustrates the process that takes place to configure, establish, and activate traffic-engineered signalled LSPs.

FIGURE 178 How traffic-engineered LSPs are configured, established, and activated



Traffic-engineered, signalled LSPs are configured, established, and activated by the following processes (but with some differences between OSPF and IS-IS):

CSPF calculates a traffic-engineered path

When you configure a signalled Label Switched Path, you specify the address of the egress LER, as well as optional attributes, such as the LSP's priority and bandwidth requirements. You can optionally specify a path of LSRs that the LSP must pass through on the way to the egress LER. When you enable the signalled LSP, the ***Constrained Shortest Path First (CSPF)*** process on the ingress LER uses this information to calculate a ***traffic-engineered path*** between the ingress and egress LERs.

CSPF is an advanced form of the Shortest Path First (SPF) process used by IGP routing protocols. The CSPF process on the ingress LER uses the configured attributes of the LSP, user-specified path (if there is one), and the information in the ***Traffic Engineering Database*** (TED) to calculate the traffic-engineered path, which consists of a sequential list of the physical interfaces that packets

assigned to this LSP will pass through to travel from the ingress LER to the egress LER. The traffic-engineered path takes into account the network topology, available resources, and user-specified constraints. The traffic-engineered path calculated by CSPF may or may not be the same as the shortest path that would normally be calculated by standard IGP routing protocols.

CSPF is enabled by default for signalled LSPs, but can be disabled. When signalled LSPs are configured without CSPF, the shortest path from the ingress LER to the egress LER is calculated using standard hop-by-hop routing methods. If the LSP also is configured to use a user-specified path, the device calculates the shortest path between each LSR in the path. As with CSPF, the output of this process is a fully specified path of physical interfaces on LSRs.

The advantage of configuring signalled LSPs without CSPF is that it can span multiple OSPF areas or IS-IS levels. Since OSPF-TE LSAs and IS-IS LSPs with TE extensions have area/level flooding scope, the information in an LSR's TED is relevant only to their area or level. Consequently, signalled LSPs that use CSPF can span only an OSPF area or IS-IS level. Signalled LSPs that do not use CSPF, because they do not rely on information in the TED, do not have this restriction.

Once the path for the LSP has been calculated, RSVP signalling then causes resources to be reserved and labels to be allocated on each LSR specified in the path. This may cause already existing, lower priority LSPs to be preempted. Once resources are reserved on all the LSRs in the path, the signalled LSP is considered to be ***activated***; that is, packets can be forwarded over it.

The following sections provide additional information about the individual components of the process for activating traffic-engineered signalled LSPs, illustrated in Figure 178 on page 1297.

OSPF-TE Link State Advertisements for MPLS interfaces

MPLS-enabled devices running OSPF can be configured to send out LSAs that have special extensions for traffic engineering. These LSAs, called ***OSPF-TE LSAs***, contain information about interfaces configured for MPLS. The OSPF-TE LSAs are flooded throughout the OSPF area. LSRs that receive the OSPF-TE LSAs place the traffic engineering information into a TED, which maintains topology data about the nodes and links in the MPLS domain.

Traffic engineering information is carried in OSPF traffic engineering (OSPF-TE) LSAs. OSPF-TE LSAs are Type 10 Opaque LSAs, as defined in RFC 2370. Type 10 Opaque LSAs have area flooding scope.

OSPF-TE LSAs have special extensions that contain information related to traffic engineering; these extensions are described in RFC 3630. The extensions consist of Type/Length/Value triplets (TLVs) containing the following information:

- Type of link (either point-to-point or multiaccess network)
- ID of the link (for point-to-point links, this is the Router ID of the LSR at the other end of the link; for multiaccess links, this is the address of the network's designated router)
- IP address of the local interface for the link
- IP address of the remote interface for the link (this could be zero for multicast links)
- Traffic engineering metric for the link (by default, this is equal to the OSPF link cost)
- Maximum bandwidth on the interface
- Maximum reservable bandwidth on the interface
- Unreserved bandwidth on the interface
- Administrative groups to which the interface belongs

When configured to do so, the device sends out OSPF-TE LSAs for each of its MPLS-enabled interfaces. You can optionally specify the maximum amount of bandwidth that can be reserved on an interface, as well as assign interfaces to administrative groups. refer to "Setting traffic engineering parameters for MPLS interfaces" on page 1323 for more information.

The following events trigger the device to send out OSPF-TE LSAs:

- Change in the interface's administrative group membership
- Change in the interface's maximum available bandwidth or maximum reservable bandwidth
- Significant change in unreserved bandwidth per priority level:
  - If for any priority level, the difference between the previously advertised unreserved bandwidth and the current unreserved bandwidth exceeds 5 percent of the maximum reservable bandwidth
  - Any changes while the total reserved bandwidth exceeds 95 percent of the maximum reservable bandwidth

In addition, OSPF-TE LSAs can be triggered by OSPF; for example, when an interface's link state is changed. When an interface is no longer enabled for MPLS, the device stops sending out OSPF-TE LSAs for the interface.

IS-IS Link State Protocol data units with TE extensions for MPLS interfaces

An MPLS-enabled device running IS-IS can be configured to send out ***Link State Protocol (LSP)*** data units that contain special extensions to support Traffic Engineering (TE). (In this section—and nowhere else in this chapter—LSP is the acronym for Link State Protocol. In other sections, LSP means Label Switched Path.) These LSPs are composed of a fixed header and a number of tuples known as Type/Length/Value triplets (TLVs). LSPs that are used for traffic engineering contain a new object called a sub-TLV. Sub-TLVs are similar to regular TLVs except that, where regular TLVs exist inside IS-IS packets, sub-TLVs reside within regular TLVs. Each sub-TLV consists of three fields: a one-octet Type field, a one-octet Length field, and zero or more octets of Value.

These LSPs are flooded throughout the IS-IS domain. LSRs that receive the IS-IS LSPs with TE extensions place the traffic engineering information into a ***Traffic Engineering Database (TED)***, which maintains topology data about the nodes and links in the MPLS domain.

IS-IS LSPs have special extensions that contain information related to traffic engineering and are described in RFC 3784. The extensions consist of Type/Length/Value triplets (sub-TLVs) containing the following information:

- IP address of the local interface for the link
- IP address of the remote interface for the link (for point-to-point adjacencies)
- Traffic engineering metric for the link (by default, this is equal to the ISIS link cost)
- Maximum bandwidth on the interface
- Maximum reservable bandwidth on the interface
- Unreserved bandwidth on the interface
- Administrative groups to which the interface belongs

When configured to do so, the device sends out IS-IS LSPs with TE extensions for each of its MPLS-enabled interfaces. You can optionally specify the maximum amount of bandwidth that can be reserved on an interface, as well as assign interfaces to administrative groups. Refer to "Setting traffic engineering parameters for MPLS interfaces" on page 1323 for more information.

Any of the following events trigger the device to send out IS-IS LSPs with a TE extension:

- Change in the interface's administrative group membership.
- Change in the interface's maximum available bandwidth or maximum reservable bandwidth.
- Significant change in unreserved bandwidth per priority level, which can be either of the following:
  - For any priority level, the difference between the previously advertised, unreserved bandwidth and the current, unreserved bandwidth exceeds 5 percent of the maximum reservable bandwidth.
  - Any change if the total reserved bandwidth exceeds 95 percent of the maximum reservable bandwidth.

In addition, IS-IS LSPs with TE extensions can be triggered by IS-IS (for example, when an interface's link state changes). Furthermore, when an interface is no longer enabled for MPLS, the device stops sending out IS-IS LSPs with TE extensions for that interface.

Traffic engineering database

An LSR TED stores topology information about the MPLS domain. This topology information comes from OSPF-TE LSAs and IS-IS LSPs with TE extensions that are flooded throughout the OSPF area or IS-IS domain. When an LSR receives OSPF-TE LSAs or IS-IS LSPs with TE extensions from neighboring LSRs, it places the traffic engineering information into its TED. In this way, each LSR in the OSPF area builds an identical topology database that reflects the traffic engineering constraints, bandwidth reservations, and administrative group memberships of the area's MPLS-enabled interfaces and the links that connect them.

The topology information in the TED is used by the CSPF process when it calculates traffic-engineered paths for signalled LSPs, as the section "How CSPF calculates a traffic-engineered path"describes. You can display the contents of an LSR's TED (refer to "Displaying MPLS and RSVP information" on page 1354).

LSP attributes and requirements used for traffic engineering

In addition to the topology information in the TED, the device considers attributes and requirements specified in configuration statements for the LSP. The following user-specified parameters are considered when the device calculates a traffic-engineered path for a signalled LSP:

- Destination address of the egress LER
- Explicit path to be used by the LSP
- Bandwidth required by the LSP
- Setup priority for the LSP
- Metric for the LSP
- Whether the LSP includes or excludes links belonging to specified administrative groups

Refer to "Configuring signalled LSP parameters" on page 1334 for more information on how to set these parameters.

How CSPF calculates a traffic-engineered path

Using information in the TED in addition to the attributes and requirements of the LSP, CSPF calculates a traffic-engineered path for the LSP by performing the tasks listed below.

1. If more than one LSP needs to be enabled, CSPF selects the LSP for path calculation based on the LSP's setup priority and bandwidth requirement.

   When multiple LSPs are enabled simultaneously, such as when the device is booted, CSPF calculates the paths one at a time. CSPF starts with the LSP that has the highest configured setup priority. If more than one LSP has the same setup priority, CSPF calculates the path first for the LSP with the highest configured bandwidth requirement.

2. Eliminate unsuitable links from consideration.

   The device examines the topology information in its TED and uses this information to eliminate links from consideration for the traffic-engineered path. A link is eliminated if any of the following are true:

   - The link is half duplex.
   - The link does not have enough reservable bandwidth to fulfill the LSP's configured requirements.
   - The LSP has an **include** statement, and the link does not belong to an administrative group in the statement.
   - The LSP has an **exclude** statement, and either the link belongs to an administrative group specified in the exclude statement or the link does not belong to any administrative group at all.

3. Using the remaining links, calculate the shortest path through the MPLS domain.

   Using the links that were not eliminated in the previous step, the device calculates the shortest path between the ingress and egress LERs. If the LSP is configured to use an explicit path, the device individually calculates the shortest path between each node in the path. Refer to "Setting up paths" on page 1333 for more information on explicit paths.

   By default, the path calculated by CSPF can consist of no more than 255 hops, including the ingress and egress LERs. You can optionally change this maximum to a lower number. Refer to "Limiting the number of hops the LSP can traverse" on page 1343.

4. If multiple paths have the same cost, select one of them.

   The shortest path calculation performed in the previous step may result in multiple, equal-cost paths to the egress LER. In this case, the device chooses the path whose final node is the physical address of the destination interface.

   If more than one path fits this description, by default, the device chooses the path with the fewest hops. If multiple paths have this number of hops, the device chooses one of these paths at random. You can optionally configure the device to choose the path that has either the highest available bandwidth or the lowest available bandwidth. Refer to "Specifying a tie-breaker for selecting CSPF equal-cost paths" on page 1344.

The output of the CSPF process is a traffic-engineered path, a sequential list of the physical interfaces that packets assigned to this LSP pass through to reach the egress LER. Once the traffic-engineered path has been determined, RSVP signalling attempts to establish the LSP on each LSR in the path. Refer to the next section, "How RSVP establishes a signalled LSP", for a description of how this works.

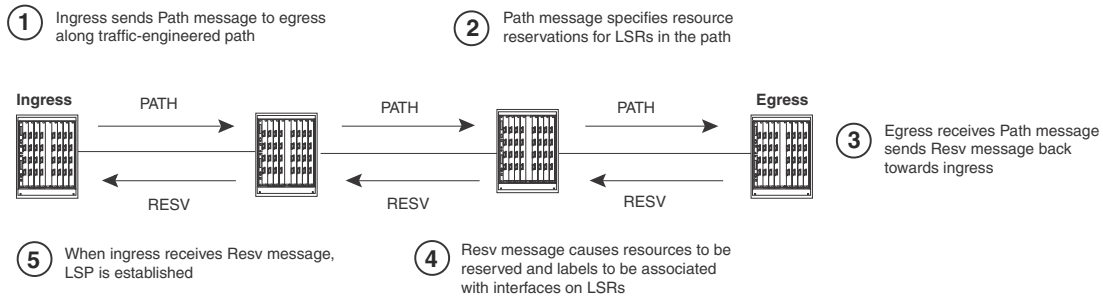
How RSVP establishes a signalled LSP

The traffic-engineered path calculated by CSPF consists of a sequential list of physical interface addresses, corresponding to a path from the ingress LER to the egress LER. Using this traffic-engineered path, RSVP establishes the forwarding state and resource reservations on each LSR in the path.

As with OSPF, special extensions for traffic engineering are defined for RSVP. These extensions include the EXPLICIT_ROUTE, LABEL_REQUEST, LABEL, and RECORD_ROUTE objects in addition to the Fixed Filter (FF) reservation style. These extensions are described in RFC 3209.

The following diagram illustrates how RSVP establishes a signalled LSP.

FIGURE 179 How RSVP establishes a signalled LSP



RSVP signalling for LSPs works as described below.

1. The ingress LER sends an RSVP Path message towards the egress LER.

   The Path message contains the traffic engineered path calculated by the CSPF process, specified as an EXPLICIT_ROUTE object (ERO). The Path message travels to the egress LER along the route specified in the ERO.

   The Path message also describes the traffic for which resources are being requested and specifies the bandwidth that needs to be reserved to accommodate this traffic. In addition, the Path message includes a LABEL_REQUEST object, which requests that labels be allocated on LSRs and tells the egress LER to place a LABEL object in the Resv message that it sends back to the ingress LER.

   Before sending the Path message, the ingress LSR performs ***admission control*** on the outbound interface, ensuring that enough bandwidth can be reserved on the interface to meet the LSP's requirements. Admission control examines the LSP's configured ***setup priority*** and ***mean-rate*** settings. For the LSP to pass admission control, the outbound interface must have reservable bandwidth at the LSP's setup priority level that is greater than the amount of bandwidth specified by the LSP's mean-rate setting. Refer to "Admission control, bandwidth allocation, and LSP preemption", for more information and examples of this process.

2. The Path message requests resource reservations on the LSRs along the path specified in the ERO.

   If the LSP passes admission control, the ingress LER sends a Path message to the address at the top of the ERO list. This is the address of a physical interface on the next LSR in the path. As the ingress LER did, this LSR performs admission control to make sure the outbound interface has enough reservable bandwidth to accommodate the LSP.

   If the LSP passes admission control, the LSR then removes its address from the top of the ERO list and sends the Path message to the address now at the top of the ERO list. This process repeats until the Path message reaches the last node in the ERO list, which is the egress LER.

3. The egress LER receives the Path message and sends a Resv message towards the ingress LER.

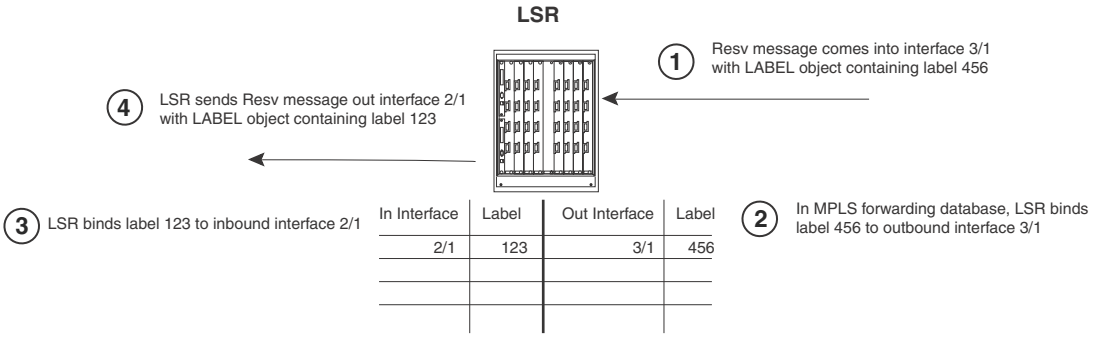
   Resv messages flow upstream from the receiver of the Path message to the sender (that is, from the egress LER to the ingress LER), taking the exact reverse of the path specified in the ERO. In response to the LABEL_REQUEST object in the Path message, the Resv message from the egress LER includes a LABEL object. The LABEL object is used to associate labels with interfaces on the LSRs that make up the LSP.

4. As the Resv messages travel upstream, resources are reserved on each LSR.

   When an LSR receives a Resv message, it again performs admission control on the interface where the Resv message was received (that is, the interface that will be the outbound interface for packets travelling through the LSP). If the LSP still passes admission control, bandwidth is allocated to the LSP. The LSR allocates the amount of bandwidth specified by the LSP's mean-rate setting, using bandwidth available to its ***hold priority*** level. This may cause lower priority LSPs active on the device to be preempted.

   Once bandwidth has been allocated to the LSP, the LABEL object in the Resv message is used to associate labels with interfaces in the LSR's MPLS forwarding table. Figure 180 shows an example of how this works.

FIGURE 180 How the RSVP LABEL object associates a label with an interface in the MPLS forwarding table



In the example above, the LSR receives a Resv message on interface 3/1 from the downstream LSR in the ERO. The Resv message has a LABEL object containing label 456. After performing admission control and bandwidth allocation, the LSR adds an entry to its MPLS forwarding table for this LSP, associating label 456 with outbound interface 3/1.

The LSR then takes a label from its range of available labels (for example, 123) and places it in the LABEL object in the Resv message that it sends to the upstream LSR. In this example, the LSR sends the Resv message out interface 2/1 to the upstream LSR in the ERO. In its MPLS forwarding table for this LSP, the LSR associates label 123 with inbound interface 2/1.

This process repeats at each LSR until the Resv message reaches the ingress LER.

NOTE
To enable penultimate hop popping for the LSP, the LABEL object sent by the egress LER to the penultimate LSR contains a value of 3 (Implicit Null Label). This is an IETF-reserved label value that indicates to the penultimate LSR that it must pop the label of MPLS-encoded packets that belong to this LSP.

5. Once the Resv message reaches the ingress LER, and the process described in Step 4 takes place, the LSP is activated. At this point each LSR in the LSP has reserved resources, allocated labels, and associated labels with interfaces. The LSP is activated, and the ingress LER can assign packets to the LSP.

Refresh messages

Once a signalled LSP is enabled at the ingress LER, the router persistently attempts to establish the LSP through periodic retries until the LSP is successfully established. To maintain the forwarding states and resource reservations on the routers in an LSP, Path and Resv messages are exchanged between neighboring LSRs at regular intervals. If these refresh messages are not received on the routers in the LSP, the RSVP forwarding states and resource reservations are removed. You can control how often the Path and Resv messages are sent, as well as how long the device waits before removing forwarding states and resource reservations. Refer to "Setting RSVP parameters" on page 1327 for more information.

Admission control, bandwidth allocation, and LSP preemption

When a Resv message is received on an LSR, admission control determines whether the LSP can be established, based on its configured priority. If an LSP passes admission control, bandwidth is allocated to the new LSP, possibly preempting existing LSPs that have lower priority.

An LSP's priority consists of a ***setup*** priority and a ***hold*** priority. The setup priority is the priority for taking resources; the hold priority is the priority for holding resources. An LSP's setup priority is considered during admission control, and its hold priority is considered when bandwidth is allocated to the LSP. The setup and hold priorities are expressed as numbers between 0 (highest priority level) and 7 (lowest priority level). An LSP's setup priority must be lower than or equal to its hold priority. You can configure either of these values for an LSP; by default, an LSP's setup priority is 7 and its hold priority is 0.

On an MPLS-enabled interface, a certain amount of bandwidth is allocated for usage by LSPs; this amount can be either the maximum available bandwidth on the interface (the default) or a user-specified portion. The amount of bandwidth an individual LSP can reserve from this pool of allocated bandwidth depends on two user-configured attributes of the LSP: the LSP's priority and the LSP's ***mean-rate*** (the average rate of packets that can go through the LSP). The following conditions also apply:

- For an LSP to pass admission control, the bandwidth available to its setup priority level must be greater than the value specified by its mean-rate.
- If an LSP passes admission control, the bandwidth specified by its mean-rate is allocated to the LSP, using bandwidth available to its hold priority level.
- For the allocation of bandwidth to the new LSP, the system might preempt existing, lower-priority LSPs.

When setting up an LSP, the device actually performs admission control twice: when the Path message is received and when the Resv message is received. If the LSP passes admission control after the Resv message is received, bandwidth allocation and LSP preemption take place.

The sections that follow include examples of how admission control, bandwidth allocation, and preemption work.

Admission control

Admission control examines the LSPs setup priority and mean-rate settings to determine whether the LSP can be activated. To pass admission control, the reservable bandwidth available at the LSP's setup priority level must be greater than the value specified by its mean-rate.

For example, if the maximum reservable bandwidth on an interface is 10,000 Kbps and no LSPs are currently active, the amount of reservable bandwidth on the interface for each priority level would be as follows:

Priority	Unreserved Bandwidth
0	10,000
1	10,000
2	10,000
3	10,000
4	10,000
5	10,000
6	10,000
7	10,000
Active LSPs: None	

The LSR receives a Resv message for an LSP that has a configured setup priority of 6 and a hold priority of 3. The mean-rate specified for this LSP is 1,000 Kbps. For priority level 6, up to 10,000 Kbps can be reserved. Because the configured mean-rate for this LSP is only 1,000 Kbps, the new LSP passes admission control.

Bandwidth allocation

Once the LSP passes admission control, bandwidth is allocated to it. The bandwidth allocation procedure examines the LSP's hold priority and mean-rate settings. The amount of bandwidth specified by the mean-rate is allocated to the LSP, using reservable bandwidth available at the LSP's hold priority level.

In this example, the LSP's hold priority is 3 and mean-rate is 1,000 Kbps. On this interface, for priority level 3, up to 10,000 Kbps can be reserved. The amount of bandwidth specified by the mean-rate (1,000 Kbps) is allocated to the LSP.

After bandwidth is allocated to this LSP, the amount of unreserved bandwidth on the interface is reduced accordingly. In the example, the reservable bandwidth array for the interface now looks like this:

Priority	Unreserved Bandwidth
0	10,000
1	10,000
2	10,000
3	9,000
Active: LSP with setup 6, hold 3, mean-rate 1,000	

Priority	Unreserved Bandwidth
4	9,000
5	9,000
6	9,000
7	9,000
Active: LSP with setup 6, hold 3, mean-rate 1,000	

Given the bandwidth allocation above, if an LSP were established with a setup priority of 3 and a mean-rate of 9,500 Kbps, it would not pass admission control because only 9,000 Kbps is available at priority 3.

LSP preemption

If there is not enough unallocated bandwidth on an interface to fulfill the requirements of a new LSP that has passed admission control, existing LSPs that have a lower priority may be preempted. When preemption occurs, bandwidth allocated to lower-priority LSPs is reallocated to the higher-priority LSP. LSP preemption depends on the bandwidth requirements and priority of the new LSP, compared to the bandwidth allocation and priority of already existing LSPs.

When LSP preemption is necessary, the device uses the following rules:

- Preempt existing LSPs that have lower priority than the new LSP.
- If several existing LSPs have lower priority than the new LSP, preempt the LSP that has the lowest priority.
- If two LSPs have equal priority and one must be preempted, preempt the LSP with the higher bandwidth requirement.
- Preempt the fewest number of LSPs necessary.

In the example above, bandwidth has been allocated to an LSP that has a hold priority of 3 and a mean-rate of 1,000 Kbps. When a new LSP with a setup priority of 2, hold priority of 1, and mean-rate of 10,000 Kbps is established, admission control, bandwidth allocation, and LSP preemption work as described below.

1. **Admission control:** On the interface, there is 10,000 Kbps available to priority 2. The mean-rate for the new LSP is 10,000, so the LSP passes admission control; bandwidth can be allocated to it.
2. **Bandwidth allocation:** The hold priority for the new LSP is 1. On the interface, 10,000 Kbps is available to priority 1. This entire amount is allocated to the LSP.
3. **LSP preemption:** The first LSP had been using 1,000 Kbps of this amount, but its hold priority is only 3. Consequently, the first LSP is preempted, and its bandwidth allocation removed in order to make room for the new LSP.

Once this happens, the reservable bandwidth array for the interface looks like this:

Priority	Unreserved Bandwidth
0	10,000
1	0
2	0
3	0
4	0
5	0
6	0
7	0
Active: LSP with setup 2, hold 1, mean-rate 10,000 **Preempted:** LSP with setup 6, hold 3, mean-rate 1,000	

On this interface, the only LSP that could preempt the active LSP would be have a setup and hold priority of 0.

When multiple LSPs are candidates for preemption, the device normally preempts the LSP with the lowest priority. However, if preempting a higher priority LSP with a high bandwidth requirement would allow lower priority LSPs with lower bandwidth requirements to avoid preemption, the higher-priority LSP is preempted.

For example, consider an interface with 10,000 Kbps of reservable bandwidth, allocated to two active LSPs: one with a setup priority of 3, hold priority of 2, and mean-rate of 5,000 Kbps; and another with a setup priority of 4, hold priority of 3, and mean-rate of 2,500 Kbps. When an LSP with a setup priority of 1, hold priority of 0, and mean-rate of 7,500 Kbps is established, the following take place.

1. **Admission control:** On the interface, there is 10,000 Kbps available to priority 1. The mean-rate for the new LSP is 7,500 Kbps, so the LSP passes admission control; bandwidth can be allocated to it.
2. **Bandwidth allocation:** The hold priority for the new LSP is 0. On the interface, 10,000 Kbps is available to priority 0. Of this amount, 7,500 Kbps is allocated to the new LSP.
3. **LSP preemption:** To reserve enough bandwidth for the new LSP, one of the active LSPs must be preempted. The LSP with hold priority 2 uses 5,000 Kbps, and the LSP with hold priority 3 uses 2,500 Kbps. Instead of preempting both LSPs, the device preempts the higher priority LSP and its allocation of 5,000 Kbps. This clears enough bandwidth to allow both the new LSP and the lower priority LSP to be active.

After preemption, the reservable bandwidth array for the interface looks like this:

Priority	Unreserved Bandwidth
0	2,500
1	2,500
2	2,500
3	0
4	0
5	0
6	0
7	0
Active: LSP with setup 1, hold 0, mean-rate 7,500 LSP with setup 4, hold 3, mean-rate 2,500 **Preempted:** LSP with setup 3, hold 2, mean-rate 5,000	

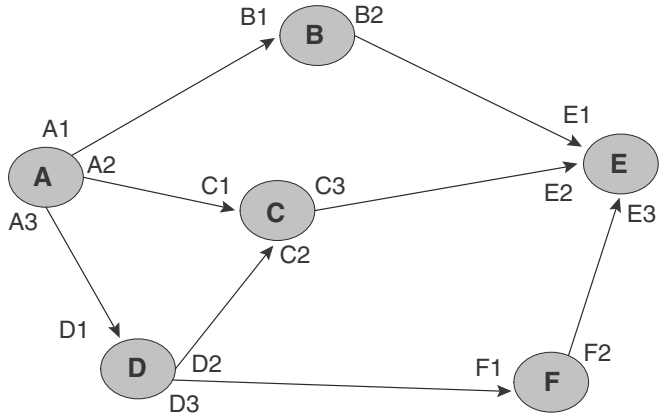
Calculating a path based on an interface address

Under normal conditions, router IDs are used to configure hops within an MPLS path. In situations where you want to exercise more control over the path, you can specify actual interface addresses in the MPLS path to make sure that the path will traverse the interface specified. In previous versions, the CSPF calculation would always resolve a specified interface address to the router ID. Consequently, although a particular interface on a router is specified, the CSPF calculation can end up connecting the path through a different interface on the router where the interface has been specified.

In the network described in Figure 181, the source node is “A” and the destination node is “E”. In this configuration, incoming and outgoing interfaces are defined in the figure by their relationship to where the arrowhead on the connecting line points. The arrowheads point to the incoming interface from the outgoing interface. For instance “A1”, “A2” and “A3” are the outgoing interfaces of node A and “C1” and “C2” are the incoming interfaces of node C. The following example describes how the router might calculate a path between “A” and “B” under the default operating condition.

In this example, an MPLS path has been configured with a source “A” and a destination “E1”. Under default operation, the interface “E1” destination is resolved to the routerID for “E”. This means that the path can be calculated to arrive at the “E” node on any of the following interfaces: “E1, “E2” or “E3”. While a path that travels from node “A” to node “B” to node “E” is the only path that actually satisfies the intent of the configuration, any of the following paths could be created by CSPF under the default operation condition: “A” to “C” to “E”, “A” to “D” to “C” to “E” or “A” to “D” to “F” to “E”.

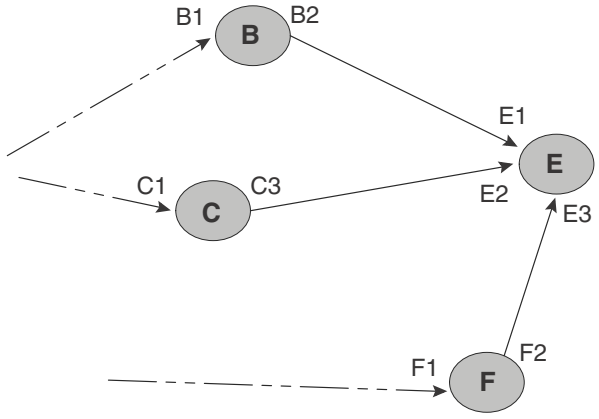
FIGURE 181 Calculating a path based on an interface



The global **cspf-interface-constraint** command directs the router to include the interface address as a constraint when it determines the shortest path. When invoked, this command ensures that a specified interface must be included in an LSP. This constraint can be turned on and off dynamically and does not affect established primary or secondary LSPs. CSPF interface constraint is significant for the ingress node only, where CSPF calculation takes place for an LSP.

When configuring CSPF interface constraint, you should be aware that the imposition of this additional constraint can increase the possibility of no path being found where otherwise there could be a path. One case where this can occur is where the path required to conform to the interface constraint fails the configured bandwidth constraint. Additionally, no path may be found where a configured path contains an inherently contradictory condition. For example if a path is configured "B1 (strict) to E2 (loose) as shown in Figure 182, no path will be found. This is because CSPF will always append B1 into the final CSPF path. This has the effect of making "B" the source node of the next hop and will therefore exclude "E1 as a traversed interface in subsequent paths to the destination node "E". Consequently, in this example the LSP will be down. However, if the **cspf-interface-constraint** command is not active, a CSPF path will be found and the LSP will go up.

FIGURE 182 Example of where no path is found



The **cspf-interface-constraint** command is described in "Configuring CSPF interface constraint" on page 1322.

MPLS fast reroute using one-to-one backup

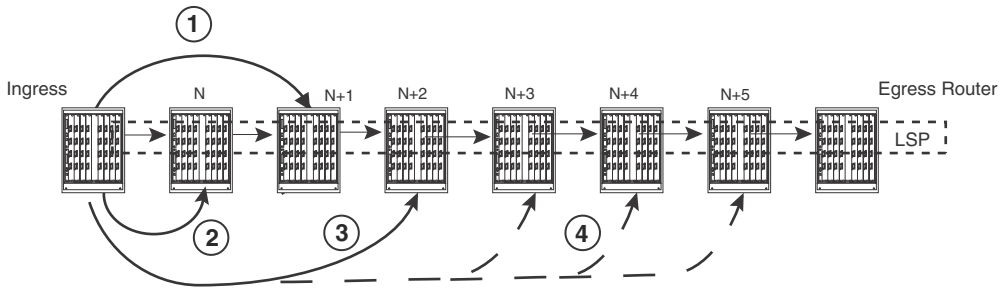
The Multi-Service IronWare software supports MPLS Fast Reroute to provide the ability for an LSP to route traffic around a failed node by using a detour route as described in RFC 4090. By using the one-to-one backup method, each LSR except the egress router is identified as a Point of Local Repair (PLR). Each PLR tries to initiate a *detour LSP* to provide a backup route for the protected path. This detour LSP is used to reroute traffic locally on the detour path in the event of a failure on the protected path.

Finding a detour at a PLR

Figure 183 illustrates how the algorithm works to determine the detour at each PLR.

NOTE
Although the example illustrates this method from only the Ingress router point-of-view, the same functionality operates on each PLR in the protected path.

FIGURE 183 Fast reroute using one-to-one backup



As shown In Figure 183, MPLS Fast Reroute operates according to the steps in the following list in a situation where the path from the ingress router to router N becomes inoperable.

1. The router first tries to find a detour path from the ingress router to the N + 1 node that excludes the failed link that the protected path traverses out of the ingress route and Node N.
2. If unable to find a detour path to node N + 1, in step 1, the router attempts to find a detour path from the ingress router to node N that excludes the link that the protected path traverses out of the ingress router.
3. If it is unable to find a detour path in steps 1 and 2, it attempts to find a detour path to any downstream node (until it reaches the egress LSR) immediately following the node N+1 in strict order. The exclusion criteria includes the downstream links (in the direction of the protected LSP) used in the protected path at each PLR.

Failover sequence

The following steps describe what happens when the ingress LER learns that a downstream break along an LSP has caused the LSP to take a detour.

1. At the PLR, the LSP's traffic has switched over to a detour within 50 msecs. Signaling has informed the router at the ingress LER of the tunnel that this event has occurred.
2. If the secondary path configured is a standby and it is in an operationally UP state, the ingress LER waits up to two minutes before switching the traffic to the LSP's secondary path. If the secondary path configured is a non-standby, the ingress LER attempts to bring the secondary path UP. Once the non-standby secondary path comes up, the ingress LER switches the traffic to the secondary path immediately.
3. The ingress LER tears down the LSP's primary path and builds a new primary path.

After the new primary path is up for the duration of the user-configured LSP revert timer, the LSP switches over to the primary LSP path.

MPLS Fast Reroute using facility backup over a bypass LSP

A bypass LSP is an MPLS LSP that serves as a tunnel to support facility backup of multiple, Fast Reroute LSPs, as specified in RFC 4090. Although the underlying mechanism of this feature is facility backup, the execution of facility backup is implemented through a user-defined bypass LSP, so this section focuses on bypass LSP.

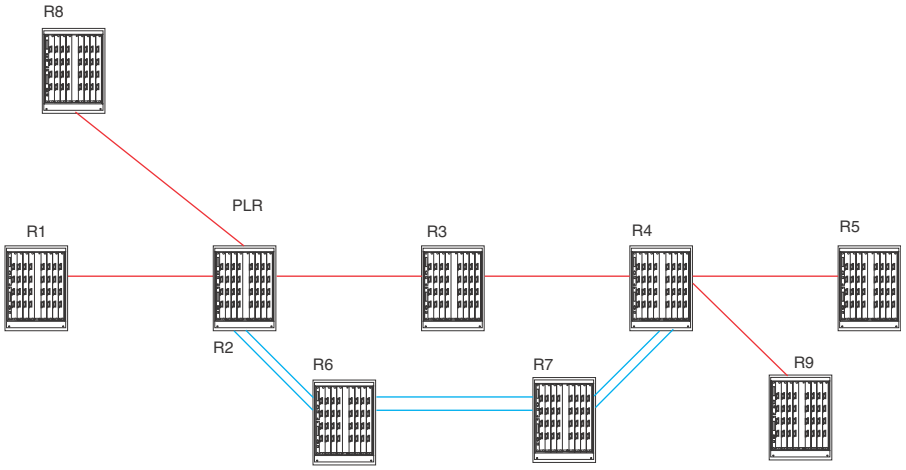
The advantage to using bypass LSP is an improvement in the scalability of protection. It provides a nearly hitless backup and, as a result, improves network resiliency. A bypass LSP consists of a predefined tunnel with a list of LSPs for which it is always ready to reroute traffic and is, therefore, a many-to-one backup. (With a detour backup, as described in "MPLS fast reroute using one-to-one backup" on page 1310, the network calculates an end-to-end detour for each disrupted LSP.)

The following definitions are important for understanding and configuring bypass LSPs:

- **Protected LSP:** An LSP whose traffic is carried over the bypass LSP if a link or router fails along the path of the protected LSP. When an MPLS LSP is configured to have Fast Reroute backup, that LSP can also be configured to request either facility backup or one-to-one backup.
- **Facility backup:** The standards-based mechanism for many-to-one backup.
- **Bypass LSP:** A tunnel that carries traffic if any number of its protected LSPs fail.
- **Point of local repair (PLR):** A router where the protected LSP and the bypass LSP first intersect and the LSP's bypass protection begins. Put another way, the PLR is the ingress of the bypass LSP. The PLR can be the ingress of the protected LSP or a transit node of the protected LSP. (refer to PLR/R2 in Figure 184 and the description in "Configuring a bypass LSP" on page 1312.)
- **Merge point (MP):** The egress router of the bypass LSP, where it merges the traffic back into the protected LSPs. (R4 in Figure 184 is the MP.) At the MP, the protected LSPs continue to carry traffic towards their own egress routers. Just as the PLR is common to all the LSPs protected by a specific bypass LSP, the MP must also be common to the protected LSPs.
- **Exclude interface:** An MPLS interface that is either a physical interface or a LAG and has the following traits:
  - It is an interface on the path of the protected LSP. (The notion that an excluded interface is protected by a bypass LSP is described in "Configuring a bypass LSP" on page 1312.)

- It is an interface that cannot be part of the bypass LSP itself.
- Exclude interfaces can consist of individual interfaces, ranges of interfaces, groups, or a LAG.

FIGURE 184 Facility backup applied to multiple routers over a bypass LSP



Configuring a protected LSP

To acquire the protection of one or more bypass LSPs along its route, an LSP that is requesting facility backup checks the interfaces that it traverses for the availability of a bypass LSPs that meet its requirements. (A Fast Reroute LSP that needs facility backup must request it. Refer to "Protecting MPLS LSPs through a bypass LSP" on page 1352 for the configuration steps.) The requesting LSP checks all of the bypass LSPs on the outbound interface of each router and selects a candidate bypass LSP that best meets its criteria so that, if the protected LSP fails, its traffic immediately switches to the bypass LSP that is upstream from the point of failure.

When an LSP is enabled for Fast Reroute, the CLI enters the configuration level for Fast Reroute, which has the option for requesting facility backup. Entering the keyword **facility-backup** in the Fast Reroute level configures the LSP to request facility backup as provided by a bypass LSP. Subsequently, for the LSP to acquire the protection of a bypass LSP, that bypass LSP must have the bandwidth, the constraints, the route (for the merge point), and other criteria that the LSP requires. Furthermore, the configuration of the bypass LSP itself must list the interface on the router where the candidate LSP and the bypass LSP first intersect. The description of linking a Fast Reroute LSP to a bypass LSP is in "Configuring a bypass LSP" on page 1312.

Configuring a bypass LSP

The crucial topics to understand for configuring a bypass LSP are the PLR, the merge point, and the excluded interfaces. This section provides a detailed definition of these items and describes how they relate to each other. Refer to Figure 184 and Figure 185 for the description of these topics.

The PLR is the ingress of a bypass LSP. If a protected link breaks downstream from the PLR, the bypass LSP carries the traffic of the LSPs it protects around the break. As shown in Figure 184, the LSPs from R1 and R8 enter R2 (the PLR). The double line that originates at R2 and then traverses R6 and R7 to terminate at R4 is the bypass LSP.

In Figure 184, the egress router for the protected LSPs is R5. Upstream from R5 is the point (R4) where the bypass LSP terminates and merges the traffic back into the protected LSPs. This router is the merge point.

In facility backup, the interfaces that go into this arrangement can belong to either:

- The bypass LSP
- The protected LSP

The specification of a bypass LSP includes manual entry of a list of interfaces at the PLR that cannot make up the bypass LSP's own route. These *excluded* interfaces are the interfaces that the protected LSPs traverse. Therefore, from the standpoint of the bypass LSP, the protected interfaces on the PLR are called *excluded* interfaces. (If the protected interfaces were included in the backup path rather than excluded from the backup path, then the interfaces would be protecting themselves—a logical contradiction.)

In facility backup, the linkage of the protected LSP to the bypass LSP is established by the following events:

- The request from an MPLS LSP for facility backup: At the ingress node (R1 in Figure 184, for example), LSP 1 is configured to request facility backup.
- The intersection of an MPLS LSP and a bypass LSP: If an LSP requesting facility backup traverses an interface on a router (R2 Figure 184) with a bypass LSP that has LSP 1's outbound interface in its user-specified list of excluded interfaces, then LSP 1 can become protected at that point, and R2 is a PLR.

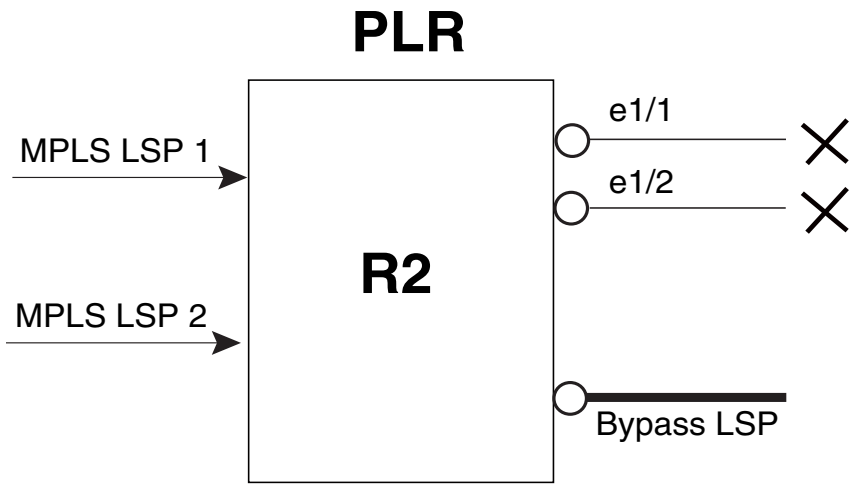
  The bypass LSP identifies the interfaces to protect in the command that creates the bypass LSP. In Figure 185, LSP 1 and LSP 2 enter R2. The outbound interfaces for these LSPs are e 1/1 and e1/2. To provide protection to LSP 1 and LSP 2, the interfaces e 1/1 and e 1/2 are listed as exclude interfaces in the configuration of the bypass LSP.

  In complex topologies, an interface can have multiple bypass LSPs protecting it. For example, the LSPs that traverse an interface might have destinations that make a single merge point impossible, so multiple bypass LSPs would be needed in this case to support different LSPs. Therefore, more than one bypass LSP can have the same interface in its list of exclude interfaces.

NOTE
The bypass LSP must have the bandwidth capacity to carry the traffic of all of its assigned LSPs. Before a candidate LSP chooses a bypass LSP on a given interface, software determines whether the bypass LSP can reserve sufficient bandwidth for the candidate LSP.

NOTE
In the current release, BFD for facility backup FRR LSP is not supported. The system returns an error if you try either to enable BFD for facility backup for an LSP or to set facility-backup mode for an LSP with BFD enabled. Further, the BFD option is not available in the bypass LSP configuration context.

FIGURE 185 Excluded interfaces on a PLR



Bypass LSP, like one-to-one backup, fits within the scope of MPLS Traffic Engineering, so the configuration of bypass LSP includes elements of traffic engineering. For example, setting up a bypass LSP relies on RSVP and CSPF. In fact, CSPF is automatically enabled on a bypass LSP and, therefore, does not appear as a configurable option at the bypass LSP configuration level.

CLI differences between a protected LSP and a bypass LSP

In the Fast Reroute context of MPLS LSP configuration, the option to request facility backup is available. For example, to request facility backup for LSP mlxe2-199, the CLI would be.

```
config-mpls-mlxe2-199-frr>#facility-backup
```

For the configuration of a bypass LSP, certain parameters are either unsupported or unnecessary. These are:

- CSPF (because it is always enabled)
- BFD (not supported in this release)
- Commit
- Secondary path
- FRR
- Selected path
- Adaptive (not supported)
- IPMTU
- Metric
- Reoptimize_timer
- Revert-timer
- Select-path
- Shortcuts

In contrast, the parameter that is unique to bypass LSP is the specification of excluded interfaces, which can be embodied as individual interfaces, ranges of interfaces, groups, or LAGs. With bypass LSP 123.

Example

```
config-mpls->#bypass-lsp 123
config-mpls-bypasslsp-123># exclude-interface <linkid>, <linkid>,
<linkid-begin-linkid-end>
```

MPLS over virtual Ethernet interfaces

Dell routers supports MPLS over virtual ethernet (VE) interfaces. MPLS over VE interfaces enables MPLS to be configured over tagged links. With this feature, MPLS can run over a single tag on the port. Other tags on the port can be used for other applications, such as Layer 2 VLANs, VPLS endpoints, VLL endpoints, etc.

An MPLS enabled VE interface supports the following services.

- IP over MPLS
- L3VPN
- Transit LSR
- PBR over MPLS
- LSP Accounting
- MPLS VLL
- MPLS VPLS
- Multicast Snooping over VPLS
- 802.1ag
- MPLS OAM
- BFD

NOTE
MPLS encapsulated packets are not supported for sFlow processing.

NOTE
Multi-port static ARP configuration in not supported for MPLS uplinks.

Configuration considerations before enabling MPLS on a VE interface

Before enabling MPLS on a VE interface, consider the configuration notes in this section.

- You must create a VE <*vid*> virtual interface id. The virtual interface id is a decimal number that represents an already configured VE interface. For more information on enabling MPLS on a VE interface, refer to "Configuring MPLS on a VE interface" on page 1328
- At least one IP address must be configured over a VE interface.
- You can enable MPLS on two or more tags on the same port.
- In the output of the **show vlan** command, MPLS packets that are received on an MPLS enabled VE interface are displayed in the Bytes received field. For more information on displaying VLAN byte counters on an MPLS enabled VE interface, refer to "Displaying VLAN information" on page 275.

In order to support configuration of MPLS uplinks and layer 2 VPN endpoints on the same physical port, consider the following:

- When an untagged or tagged Layer 2 VPN endpoint is configured and the port belongs to a MPLS VE enabled default VLAN, the configuration will be rejected. The following error message is displayed.

```
NetIron(config-mpls-vll-test)#untagged ethernet 4/3
Error - Cannot configure VLL endpoint on port 4/3 since it belongs to the
MPLS VE enabled default VLAN
```

- When an untagged or tagged Layer 2 VPN endpoint is deleted and the port is returned to the default VLAN, if an MPLS VE exists on the default VLAN, the port will automatically be converted to an MPLS uplink.

MPLS enabled interface

When enabling MPLS on a VE interface, consider the following.

- You cannot delete a VE interface while MPLS is enabled on it. You must first remove MPLS from the interface configuration. The following error message is displayed.

```
NetIron(config)#no interface ve 20
Error - VE 20 has MPLS enabled
```

- You cannot delete a VLAN associated with a VE if MPLS is enabled on that VE. You must first disable MPLS from the VE interface. The following error message is displayed:

```
NetIron(config)#no vlan 20
Error - vlan can't be deleted as MPLS is enabled on associated VE interface
```

- When MPLS is enabled on an interface, the last IP address of a VE cannot be removed. The command is rejected. The following error message is displayed:

```
NetIron(config-vif-54)#no ip address 40.40.40.5/24
IP/Port: Errno(31) Can not remove IP address as MPLS is configured on the
port
```

VPLS CPU protection

When enabling MPLS on a VE interface with VPLS CPU protection turned on, consider the following.

- VPLS CPU protection must be disabled globally, or disabled on all instances of VPLS that has VE member port as the VPLS endpoint. If VPLS CPU protection is not disabled, then MPLS cannot be enabled on a VE interface. The following error message is displayed.

```
NetIron(config-mpls)#mpls-interface ve 1
Error - Port 4/3 belongs to a VPLS instance that has CPU-protection ON
```

- You cannot configure a VPLS endpoint on a member of a MPLS VE enabled interface when VPLS CPU protection is configured globally, or for a specified instance. The configuration is rejected, and the following error message is displayed.

```
NetIron(config-mpls-vpls-test-vlan-11)#tagged ethernet 4/3
Error - VPLS instance test has CPU protection ON and port 4/3 belongs to a
MPLS VE
```

- If a VPLS endpoint belonging to a VPLS instance (with CPU protection turned on) is on a port that does not belong to the VE, then you cannot add a port in an untagged or tagged mode to a VLAN which has a MPLS VE on it.

```
NetIron(config-vlan-100)#tagged ethernet 4/7
Error - Port 4/7 belongs to a VPLS instance that has CPU protection ON
```

- On an MPLS VE enabled interface, if a VPLS endpoint is a member of the VE interface, but VPLS CPU protection is not configured for the VPLS instance, then configuring VPLS CPU protection globally will not enable CPU protection for that instance. If VPLS CPU protection is enabled locally on that instance, the configuration will also be rejected. The following error message is displayed.

```
NetIron(config-mpls)#vpls-cpu-protection
Error - Cannot configure CPU protection for VPLS 111 as end-points share
the same physical port as MPLS VE interfaces.
CPU protection feature is not turned on for VPLS 111
```

Reverse path forwarding

When enabling MPLS on a VE interface with reverse path forwarding, consider the following.

- You cannot configure MPLS on a VE interface that has at least one member port enabled with RPF strict mode. The command is rejected, and the following error message is displayed.

```
NetIron(config-if-e1000-4/3)#mpls-interface ve 1
Error - Cannot configure MPLS on VE with RPF strict mode port e 4/3
```

- You cannot configure RPF strict mode on a port that is a member of a MPLS VE interface. The command is rejected, and the following error message is displayed.

```
NetIron(config-if-e1000-4/3)#rpf-mode strict
Error: RPF: Cannot configure RPF strict mode on an MPLS VE enabled
interface
```

- You cannot add a port to a MPLS VE enabled VLAN if the RPF strict mode is already enabled on the port. The command is rejected, and the following error message is displayed.

```
NetIron(config-vlan-100)#tagged ethernet 4/3
Error - Cannot add RPF strict mode port 4/3 to MPLS VE enabled VLAN 100
```

Port mirroring

When enabling MPLS on a VE interface with port mirroring configured, consider the following.

- You cannot configure MPLS on a VE interface that has at least one member port enabled with port mirroring. The command is rejected, and the following error message is displayed

```
NetIron(config-mpls)#mpls-interface ve 54
Error - Can not configure MPLS tunnel on ve 54 with mirror port e4/3
```

- You cannot configure a MPLS VE member port as a mirror port. The command is rejected, and the following error message is displayed.

```
NetIron(config)#mirror-port e 4/3
Error: Cannot mirror a port that has MPLS VE configured
```

- You cannot add a mirror port to a MPLS VE enabled VLAN. The command is rejected, and the following error message is displayed.

```
NetIron(config-vlan-100)#tagged ethernet 4/3
Error - Cannot add mirror port 4/3 to MPLS VE enabled VLAN 100
```

Protocol-based VLANs

When enabling MPLS on a VE interface associated with a protocol-based VLAN, consider the following.

NOTE
MPLS is supported only on VE interfaces that are configured on port-based VLANs.

- You cannot configure MPLS on a VE interface associated with a protocol based VLAN. The command is rejected, and the following error message is displayed:

```
NetIron(config-mpls)#mpls-interface ve 1
Error: Cannot configure MPLS on VE built on protocol-based VLAN
```

VRF

When enabling MPLS on a VE interface for a VRF instance, consider the following.

NOTE
When configuring MPLS on a VE interface on a VLAN port, you can also configure a VRF instance on other VLANs of the same port.

- You cannot configure a VRF instance on a MPLS VE enabled interface. The following error message is displayed:

```
NetIron(config-vif-20)#vrf forwarding test
Error - cannot configure VRF on an MPLS VE enabled interface
```

Class of Service (CoS)

By default, the internal priority of a packet received on a tagged MPLS uplink is mapped from PCP and EXP bits. When determining the internal priority, the first step is to merge the PCP and EXP bits. In this step, when configuring the **qos exp force** command on an interface, the internal priority is mapped only from EXP bits and PCP bits are ignored.The **qos exp force** command does not override the port priority command. In the second step, the port priority will be merged with the internal priority and hence, the **qos exp force** command has no effect on this step.

By default, the internal priority of a packet sent out on a tagged MPLS uplink is mapped into EXP and PCP bits. When configuring the **qos pcp-encode policy off** command on an outgoing interface, the PCP bits is 0.

Configuring MPLS

This section explains how to set up MPLS on devices. It contains the following topics:

- "Enabling MPLS" on page 1319
- "RSVP message authentication" on page 1327
- "Configuring MPLS on a VE interface" on page 1328
- "Setting up signalled LSPs" on page 1333
- "Configuring signalled LSP parameters" on page 1334
- "Configuring an adaptive LSP" on page 1347

- "Configuring MPLS Fast Reroute using one-to-one backup" on page 1350
- "Protecting MPLS LSPs through a bypass LSP" on page 1352

Enabling MPLS

MPLS is disabled by default. To enable MPLS on a device, you must perform the steps listed below.

1. Enable MPLS on the device
2. Enable MPLS on individual interfaces
3. Set global MPLS policy parameters (optional)
4. Set traffic engineering parameters for MPLS-enabled interfaces (optional)
5. Set RSVP parameters (optional)

Enabling MPLS on the device

To enable MPLS on the device, enter the following commands.

```
NetIron> enable
NetIron# configure terminal
NetIron(config)# router mpls
```

Syntax: [no] router mpls

To disable MPLS on the device, use the **no** form of the command.

Enabling MPLS on individual interfaces

After you enable MPLS globally on the device, you can enable it on one or more interfaces. For example, to enable MPLS on interface e 3/1.

```
NetIron(config-mpls)# mpls-interface e 3/1
```

Syntax: [no] mpls-interface all-ethernet | ethernet <*slot/port*> **| pos** <*slot/port*> **|ve** <*vid*>

The **all-ethernet** option specifies all Layer-3 Ethernet interfaces

The **ethernet** option specifies the individual Ethernet interface described by the <*slot/port*> variable.

The **pos** option specifies the individual POS interface described by the <*slot/port*> variable.

The **ve** option specifies the individual virtual ethernet (VE) interface described by the <*vid*> variable.

Configuration Considerations for enabling MPLS on a LAG interface

When MPLS is globally enabled on the device, a port that is configured in a LAG can be enabled as an MPLS interface port to create an MPLS LAG. You can do this through either of the following approaches:

- Include a primary LAG port that has already been MPLS -enabled in a new LAG.
- MPLS-enable a primary LAG port of a previously configured LAG.

You must consider the following points when configuring MPLS on a LAG

- MPLS configuration on dynamic lag interfaces are supported.
- Switch and LACP LAGs are not supported.
- MPLS is enabled on the primary port of the LAG and this enables MPLS on the entire LAG. Secondary ports of the LAG cannot be individually configured for MPLS.

Setting global MPLS policy parameters

You can optionally set the following global MPLS policy parameters (they apply to all MPLS-enabled interfaces on the device):

- Retry time
- Retry limit
- Administrative group names
- Whether the device sends out OSPF-TE LSAs for its MPLS-enabled interfaces
- Whether the device sends out IS-IS LSPs with TE extensions for its MPLS-enabled interfaces
- Configuring IP-over-MPLS TTL Propagation Control
- LSP Accounting

Setting the retry time

When a signalled LSP is enabled, the ingress LER attempts to connect to the egress LER over the primary path specified in the LSP's configuration. If the connection is not successful, by default the ingress LER waits 30 seconds before attempting the connection again. You can configure the amount of time the ingress LER waits between connection attempts.

For example, to specify a retry time of 45 seconds.

```
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# retry-time 45
```

Syntax: [no] retry-time <*seconds*>

Setting the retry limit

If the ingress LER fails to connect to the egress LER in a signalled LSP, the ingress LER tries indefinitely to make the connection unless you set a limit for these connection attempts. After this limit is exceeded, the ingress LER stops trying to connect to the egress LER over the primary path.

If a secondary path is configured for the LSP, it is immediately activated after the primary path fails. After the secondary path is activated, the ingress LER continues to try to connect to the egress LER over the primary path either up to the configured retry limit or indefinitely if no retry limit is set. If a connection over the primary path can be established, the secondary path is deactivated, and traffic for the LSP is again sent over the primary path.

To set the number of connection attempts to 20.

```
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# retry-limit 20
```

Syntax: [no] retry-limit <*number*>

Once the connection is established, the retry counter is reset to zero. In the example above, if an LSP needs to be established again, the ingress LER will make 20 attempts to establish a connection to the egress LER.

Establishing administrative group names

Administrative groups, also known as resource classes or link colors, allow you to assign MPLS-enabled interfaces to various classes. When a device calculates the path for an LSP, it can take into account the administrative group to which a interface belongs; you can specify which administrative groups the device can include or exclude when making its calculation.

Up to 32 administrative groups can be configured on the device. You can see an administrative group either by its name or its number. Before you can see an administrative group by its name, you must specify a name for the group at the MPLS policy level and associate the name with that administrative group's number.

For example, the following commands establish three administrative group names.

```
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# admin-group gold 30
NetIron(config-mpls-policy)# admin-group silver 20
NetIron(config-mpls-policy)# admin-group bronze 10
```

Syntax: [no] admin-group <*name*> <*number*>

The <*number*> has a range of 0 – 31.

After you associate an administrative group name with a number, you can see it by name when assigning interfaces to the group or including or excluding the group from LSP calculations. Refer to "Adding interfaces to administrative groups" on page 1324 and "Including or excluding administrative groups from LSP calculations" on page 1343.

Enabling OSPF-TE LSAs for MPLS interfaces

Information related to traffic engineering is carried in OSPF traffic engineering (OSPF-TE) LSAs. OSPF-TE LSAs have special extensions that contain information about an interface's traffic engineering metric, bandwidth reservations, and administrative group memberships.

When an MPLS-enabled device receives an OSPF-TE LSA, it stores the traffic engineering information in its Traffic Engineering database (TED). The device uses information in the TED when performing calculations to determine a path for an LSP.

You can configure the device to send out OSPF-TE LSAs for all of its MPLS-enabled interfaces. To do this, enter the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# traffic-engineering ospf
```

Syntax: [no] traffic-engineering ospf [area <*area-id*> **]**

You can use the **area** option to limit the CSPF calculations to the OSPF Area specified by the <*area-id*> variable. The <area-id> variable can accept area-id in both Decimal and IP address formats.

By default, the device does not send out OSPF-TE LSAs for its MPLS-enabled interfaces. Because information in the TED is used to make path selections using CSPF and information in the TED comes from OSPF-TE LSAs or IS-IS TE LSP, you must enable the device to send out OSPF-TE LSAs or IS-IS LSPs with TE extensions if you want CSPF to perform constraint-based path selection.

The **no** option removes an existing OSPF TE database. If you use the **no** option with the **area** option, the OSPF TE database is removed for only the specified OSPF area.

Refer to "Setting traffic engineering parameters for MPLS interfaces" on page 1323, for information on the traffic engineering information carried in OSPF-TE LSAs.

Enabling IS-IS LSPs with TE extensions for MPLS interfaces

Information related to traffic engineering is carried in IS-IS traffic engineering LSPs. IS-IS TE LSPs have special extensions that contain information about an interface's administrative group memberships, IPv4 interface address, IPv4 neighbor address, maximum link bandwidth, reservable link bandwidth, unreserved bandwidth, and default traffic engineering metrics.

When an MPLS-enabled device receives an IS-IS TE LSP, it stores the traffic engineering information in its Traffic Engineering database (TED). The device uses information in the TED when performing calculations to determine a path for an LSP.

You can configure the device to send out IS-IS TE LSPs for all of its MPLS-enabled interfaces. To do this, enter the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# traffic-engineering isis level-1
```

Syntax: [no] traffic-engineering isis level-1 | level-2

The level-1 option enables LSPs with TE extensions for the IS-IS level-1 domain.

The level-2 option enables LSPs with TE extensions for the IS-IS level-2 domain.

By default, the device does not send out IS-IS LSPs with TE extensions for its MPLS-enabled interfaces. Since information in the TED is used to make path selections using CSPF, and information in the TED comes from OSPF-TE LSAs or IS-IS LSPs with TE extensions, you must enable the device to send out OSPF-TE LSAs or IS-IS LSPs with TE extensions if you want CSPF to perform constraint-based path selection.

Refer to "Setting traffic engineering parameters for MPLS interfaces" on page 1323, for information on the traffic engineering information carried in IS-IS LSPs with TE extensions.

Configuring CSPF interface constraint

As described in detail in "Calculating a path based on an interface address" on page 1308, under the default condition, hops configured as interface addresses in an LSP path are resolved to the router ID. Consequently, an LSP can be configured that does not traverse a specified interface. The **cspf-interface-constraint** command was introduced that forces the CSPF calculation to include any specified interface when creating an LSP. The operation and constraints of using this command are described in the section mentioned.

You can configure a PowerConnect router to always include a specified interface when forming an LSP by configuring the **cspf-interface-constraint** command as shown in the following.

```
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# cspf-interface-constraint
```

Syntax: [no] cspf-interface-constraint

The default condition is for the CSPF interface Constraint feature to be disabled. If the feature has been enabled, you can use the **no** option to disable it.

The CSPF interface Constraint feature may be dynamically turned on or off. Turning the feature off or on has no effect on LSPs that have already been established (primary and secondary). For LSPs that are currently retried, changing the constraint setting will change the behavior on the next retry such as when an LSP whose path is configured to use that interface fails to come up due to an interface down condition.

Also note that the CSPF interface Constraint feature has significance for the ingress node only, where the CSPF calculation takes place for an LSP or a detour segment.

Setting traffic engineering parameters for MPLS interfaces

When using constraints to determine a path for an LSP, the device takes into account information included in OSPF-TE LSAs or IS-IS LSPs with TE extensions. This information can be used to set up a path for a new LSP or to preempt an existing LSP so that an LSP with a higher priority can be established.

OSPF-TE LSAs and IS-IS LSPs with TE extensions include Type/Length/Value triplets (TLVs) containing the following information:

- Link type (either point-to-point or multiaccess network) (OSPF-TE LSAs only)
- Link ID (for point-to-point links, this is the Router ID of the LSR at the other end of the link; for multiaccess links, this is the address of the network's designated router) (OSPF-TE LSAs only)
- IP address of the local interface
- IP address of the remote interface (must exist with point-to-point links)
- Traffic engineering metric for the link
- Maximum bandwidth on the interface
- Maximum reservable bandwidth on the interface
- Unreserved bandwidth on the interface
- Administrative groups to which the interface belongs

When configured to do so with the **traffic-engineering ospf** command, the device sends out OSPF-TE LSAs containing this information for each of its MPLS-enabled interfaces. When configured to do so with the **traffic-engineering isis** command, the device sends out IS-IS LSPs containing this TE information for each of its MPLS-enabled interfaces. Optionally, you can specify the maximum amount of bandwidth that can be reserved on an interface. In addition, you can assign interfaces to administrative groups.

Reserving bandwidth on an interface

OSPF-TE LSAs and IS-IS LSPs with TE extensions contain three TLVs related to bandwidth reservation:

- The Maximum Bandwidth TLV indicates the maximum outbound bandwidth that can be used on the interface. Maximum Bandwidth is the operating speed of the port. When calculated for a LAG, the Maximum Bandwidth is the operating speed of the primary port multiplied by the number of active ports in the LAG. Hence, this reflects the actual physical bandwidth of the interface. This TLV is not configurable by the user.
- The Maximum Reservable Bandwidth TLV indicates the maximum bandwidth that can be reserved on the interface. By default, the Maximum Reservable Bandwidth is the same as the Maximum Bandwidth for the interface. You can optionally change the reservable bandwidth to an amount greater or less than the maximum available bandwidth of the interface. When a Maximum Reservable Bandwidth is configured on the primary port within a LAG, the value configured applies to the entire LAG regardless of any change to the number of active ports within the LAG. By default, the Maximum Reservable Bandwidth for the LAG is the same as its Maximum Bandwidth.
- The Unreserved Bandwidth TLV indicates the amount of bandwidth not yet reserved on the interface. This TLV consists of eight octets, indicating the amount of unreserved bandwidth (in kbits second) at each of eight priority levels. The octets correspond to the bandwidth that can be reserved with a hold priority of 0 through 7, arranged in increasing order, with priority 0

occurring at the start of the TLV, and priority 7 at the end of the TLV. The value in each of the octets is less than or equal to the maximum reservable bandwidth. The Unreserved Bandwidth TLV itself is not user-configurable, although it is affected by modifications to the reservable bandwidth on an interface, as well as changes to LSPs.

You can optionally change the amount of reservable bandwidth on an MPLS-enabled interface (that is, modify the value in the Maximum Reservable Bandwidth TLV in OSPF-TE LSAs or IS-IS TE LSPs sent out for the interface). To do this, enter commands such as the following.

```
NetIron(config-mpls)# mpls-interface e 3/1
NetIron(config-mpls-interface)# reservable-bw 10000
```

Syntax: [no] reservable-bw <*number*>

The reservable bandwidth is expressed in Kbits/sec. By default, the reservable bandwidth is the same as the maximum available bandwidth on the interface. If the amount of reservable bandwidth is greater than the maximum available bandwidth, then the link can be oversubscribed. If the reservable bandwidth is less than the maximum available bandwidth, then LSPs cannot reserve all physical bandwidth on the interface. If the **reservable-bw** command is applied to the primary port within a LAG, the bandwidth configured for that port will apply to the entire LAG regardless of any change to the number of active ports within the LAG.

Changing the amount of reservable bandwidth on an interface causes the amount of unreserved bandwidth to be recalculated. In addition, it may cause an OSPF-TE LSA or IS-IS TE LSP to be issued, as well as possibly pre-empt existing LSPs if bandwidth reservations can no longer accommodate them.

Adding interfaces to administrative groups

You can place individual interfaces into administrative groups. Administrative groups, also known as resource classes or link colors, allow you to assign MPLS-enabled interfaces to various classes. For example, you can define a group called “gold” and assign high-bandwidth interfaces to it. When a device calculates the path for an LSP, it can take into account the administrative group to which a interface belongs. You can configure up to 32 administrative groups. By default, an interface does not belong to any administrative groups.

Administrative groups are in the range 0 – 31. You can see an administrative group either by name or number. To see an administrative group by name, first create a name for the group and associate the name with an administrative group number. Refer to “Establishing administrative group names” on page 1321 for details.

To assign MPLS-enabled interface e 3/1 to an administrative group called “gold”, enter the following.

```
NetIron(config-mpls)# mpls-interface e 3/1
NetIron(config-mpls-interface)# admin-group gold
```

Syntax: [no] admin-group <*number*> | <*name*> **...**

The <*number*> can be from 0 – 31. The administrative group name <*name*> must have been previously configured.

An MPLS-enabled interface can belong to any number of administrative groups. For example, to assign an interface to group “gold” and group 31, enter commands such as the following.

```
NetIron(config-mpls)# mpls-interface e 3/1
NetIron(config-mpls-interface)# admin-group gold 31
```

After you add interfaces to administrative groups, you can specify which groups can be included or excluded from LSP calculations. Refer to "Including or excluding administrative groups from LSP calculations" on page 1343.

IP-over-MPLS TTL propagation control

In the MPLS label header, the TTL field indicates the Time To Live (TTL) value for an MPLS packet. For IP-over-MPLS applications, at the ingress LER an IP packet's TTL value is decremented by one and the IP checksum is recalculated. The IP packet's TTL value is then copied to its MPLS TTL field. At each transit LSR hop, the MPLS TTL value is decremented by 1. If the MPLS TTL value reaches 1 or 0, the packet is discarded.

At the MPLS router that pops the label (either the penultimate LSR or the egress LER), the incoming packet's MPLS TTL value is copied to the packet's IP TTL field, the IP TTL field is decremented by 1, and the checksum is recalculated. The result is that each LSR in the MPLS domain is counted as one hop. This is the default behavior.

Optionally, you can configure TTL propagation so that the entire MPLS domain appears as a two hops. In this case, the ingress LER decrements the IP packet's TTL value by one and then places a value of 255 in the packet's MPLS TTL field. The MPLS TTL value is decremented by 1 as the MPLS packet passes through each LSR in the MPLS domain. When the label is popped, the value in the MPLS TTL field is discarded, not copied to the packet's IP TTL field. The unlabeled IP packet's TTL is then decremented by one as it passes through the egress LER. This means that the packet's IP TTL is decremented twice from the time it enters the ingress LER to the time it exits the egress LER, making the MPLS domain appear as two hops.

To configure TTL propagation so that the entire MPLS domain appears as two hops, enter the following commands on both the ingress LER and the MPLS router that pops the label (either the penultimate LSR or the egress LER).

```
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# no propagate-ttl
```

Syntax: [no] propagate-ttl

When **no propagate-ttl** is configured, the ingress LER places a value of 255 into the packet's MPLS TTL field, regardless of the TTL value in the packet's IP header. The packet's IP TTL value is decremented twice: once at the ingress LER and once at the egress LER. With this option, the entire MPLS domain (regardless of the number of transit LSR hops) counts as two hops for the IP TTL value.

NOTE

If you choose to configure TTL propagation in this way, it is important that you enter the **no propagate-ttl** command at ***both*** the ingress LER and the MPLS router that pops the label. If you omit the **no propagate-ttl** command at the MPLS router that pops the label, the value in the packet's MPLS TTL field would be copied into the packet's IP TTL field. This value could be as high as 255.

Enabling LSP accounting

The LSP accounting feature provides the ability to count the number of traffic bytes and packets forwarded through a specified LSP. The LSP accounting feature is supported for the following:

- RSVP-signalled LSPs
- LDP signalled LSPs

When the command **vlan-counter exclude-overhead** is configured or removed, the LSP counters in software and hardware are flushed, and accounting starts fresh. In summary:

- **Ingress-tunnel accounting without exclude-ethernet-overhead:**
  - With **vlan-counter exclude-overhead** not configured: the size includes 20-byte Ethernet overhead (IFG+ Preamble) and 4-byte CRC.
  - With **vlan-counter exclude-overhead** configured: excludes 20-byte per-packet Ethernet overhead from byte counting.
- **Ingress-tunnel accounting with exclude-ethernet-overhead:**
  - The **exclude-ethernet-overhead** option, lets you exclude the Ethernet header (14 bytes) and Ethernet overhead (20 bytes) and CRC overhead (4 bytes) when collecting the byte statistics. In other words, it counts only the size of MPLS packet. The **exclude-ethernet-overhead** option does not work with untagged ports carrying q-in-q packets for IP over MPLS, nor does it count multiple tags in a packet.

NOTE
This feature is applicable only on LSPs for which the devices are an ingress LER.

NOTE
LSP tunnel statistics are not supported when an ingress LER is a two node LSP (PHP), or when traffic is forwarded to a directly connected Provider Edge Router (PE).

LSP accounting is disabled by default. To enable LSP accounting for an LSP, enter the following commands.

```
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# ingress-tunnel-accounting
```

Syntax: [no] ingress-tunnel-accounting [exclude-ethernet-overhead]

NOTE
The command **no ingress-tunnel-accounting exclude-ethernet-overhead** disables only the **exclude-ethernet-overhead** option. To disable **ingress-tunnel-accounting** itself, enter the command **no ingress-tunnel-accounting**.

To use this feature, you must specify an LSP accounting CAM sub-partition value by using the following sequence.

```
NetIron(config)# system-max lsp-out-acl-cam 1000
```

Syntax: [no] system-max lsp-out-acl-cam <*number*>

The <*number*> variable is the number of CAM entries available for LSP accounting. The default value is 0.

The load-interval command can be set to a time during which the average byte and packet rates are calculated as shown in the following.

```
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# load-interval 30
```

Syntax: [no] load-interval <*seconds*>

The <*seconds*> variable can be configured in multiples of 30 seconds within the range of 30 to 300 seconds. The default value of load-interval is 300 seconds.

SNMP agent support for ACL accounting

The SNMP agent supports the LSP tunnel byte count in MIB ifHCOutOctets, and LSP tunnel packet count in MIB ifHCOutUcastPkts within the ifXTable.

Setting RSVP parameters

RSVP is automatically enabled when MPLS is enabled on the device. You can optionally configure the following RSVP parameters:

- Refresh interval
- Refresh multiple

Setting the refresh interval

To maintain path states and resource reservations on the routers in an LSP, RSVP Path and Resv messages are sent at regular intervals. Path messages flow downstream in an LSP, from the ingress LER towards the egress LER. Resv messages flow upstream, in the reverse direction of Path messages.

You can control how often the Path and Resv messages are sent by setting the refresh interval. By default, the refresh interval is 30 seconds. You can set the refresh interval to between 0 – 2147483 seconds.

To set the refresh interval to 20 seconds.

```
NetIron(config-mpls)# rsvp
NetIron(config-mpls-rsvp)# refresh-interval 20
```

Syntax: [no] refresh-interval <*seconds*>

Setting the refresh multiple

If refresh messages are not received, RSVP path states and resource reservations are removed from the routers in an LSP. By default, the device waits the length of 3 refresh intervals; if no refresh message is received by the end of that time, the path state or resource reservation is removed.

The refresh multiple is the number of refresh intervals that must elapse without a refresh message before a path state or resource reservation times out. By default, the refresh multiple is 3 intervals. You can set the refresh multiple to between 0 – 65535 intervals.

To set the refresh multiple to 5 intervals.

```
NetIron(config-mpls)# rsvp
NetIron(config-mpls-rsvp)# refresh-multiple 5
```

Syntax: [no] refresh-multiple <*intervals*>

RSVP message authentication

Support was added for RSVP message authentication using MD5 as described in RFC 2747. It is implemented on the PowerConnect routers to prevent spoofing of RSVP messages. This RFC defines the use of a message digest carried in the RSVP INTEGRITY object. This object carries the following information:

- Key ID: a 8-bit number unique to a given sender
- Sequence Number: a 64-bit monotonically increasing sequence number
- Keyed Message Digest: As implemented here using MD5 it is a 16-bit message digest

In order to support RFC 2747, this implementation supports the following:

- An authentication type using the MD5 cryptographic algorithm

- An authentication key for use with the authentication algorithm
- An authentication window of 1 which specifies that the maximum number of authenticated messages that can be received out of order is 1.

Configuring RSVP message authentication

RSVP Message Authentication is disabled by default. This authentication method uses MD5 and is configured within the MPLS configuration mode.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface ethernet 1/1
NetIron(config-mpls-if-e100-1/1)# rsvp-authentication key administrator
```

Syntax: [no] rsvp-authentication key <*string*>

The <*string*> variable specifies a text string of up to 64 characters that is encrypted and used for RSVP message authentication.

By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **key** and <*string*>.

Example

```
NetIron(config-mpls-if-e100-1/1)# rsvp-authentication key 0 administrator
```

The software adds a prefix to the authentication key in the configuration. For example, the following portion of the code has the encrypted code “2”.

```
rsvp-authentication 2 key $IUA2PWc9LW9VIW9zVQ=="
```

The prefix can be one of the following:

- 0 = the key string is not encrypted and is in clear text
- 1 = the key string uses proprietary simple crytographic 2-way algorithm
- 2 = the key string uses proprietary base64 crytographic 2-way algorithm

Configuring MPLS on a VE interface

To enable MPLS on a VE interface, first create a VE interface and configure an IP address for the VE as shown in the example below.

```
NetIron(config)#vlan 100
NetIron(config-vlan-100)#tagged ethernet 2/1
NetIron(config-vlan-100)#router-interface ve 100
NetIron(config-vlan-100)#exit
NetIron(config)#interface ve 100
NetIron(config-vif-100)#ip address 10.10.10.1/24
NetIron(config-vif-100)#exit
```

Then enable MPLS on the VE interface as shown in the example below.

```
NetIron(config)#router mpls
NetIron(config-mpls)#mpls-interface ve 100
NetIron(config-mpls-if-ve-100)#
```

Syntax: [no] mpls-interface [ve <*ve-id*>]

The **mpls-interface ve** parameter allows you to enable MPLS on a VE interface. The <*ve-id*> variable allows you to specify a VE interface ID. The **no mpls-interface ve** command removes all configuration for MPLS on a VE enabled interface.

The following MPLS commands are available on a VE interface under the mpls interface configuration mode.

- admin-group - "Adding an MPLS VE interface to an administrative group" on page 1329
- ldp-enable - "Configuring LDP on an MPLS VE interface" on page 1329
- ldp-params - "Setting the LDP hello interval on an MPLS VE interface (link Only)" on page 1330
- hello-interval - "Setting the LDP hello interval on an MPLS VE interface (link Only)" on page 1330
- hello-timeout - "Setting the LDP hello holdtime on an MPLS VE interface (link only)" on page 1330
- reservable-bandwidth - "Bandwidth computation for an MPLS VE interface" on page 1331
- rsvp-authentication - "Configuring RSVP message authentication on an MPLS VE interface" on page 1332
- exclude-interface - "Specifying a bypass LSP for an MPLS VE interface" on page 1332

Adding an MPLS VE interface to an administrative group

You can place individual interfaces into administrative groups. Administrative groups, also known as resource classes or link colors, allow you to assign MPLS enabled VE interface to various classes. For more information on assigning an MPLS-enabled interface to an administrative group, refer to **"Adding interfaces to administrative groups"** on page 1324.

To assign MPLS interface ve 100 to an administrative group called "gold", enter the following.

```
NetIron(config-mpls)#mpls-interface ve 100
NetIron(config-mpls-if-ve-100)# admin-group gold
```

Syntax: [no] admin-group <*number*> | <*name*> [<*number*> | <*name*>]

The <*number*> variable can be from 0 – 31. The administrative group <*name*> variable must have been previously configured. By default, no admin group is configured for any MPLS interfaces, including MPLS VE interface.

An MPLS enabled VE interface can belong to any number of administrative groups. For example, to assign an MPLS interface ve 100 to group "gold" and group 31, enter commands such as the following.

```
NetIron(config-mpls)#mpls-interface ve 100
NetIron(config-mpls-if-ve-100)# admin-group gold 31
```

Configuring LDP on an MPLS VE interface

NOTE
For more information on configuring LDP on physical interfaces, refer to "Configuring LDP on an interface" on page 1412.

To configure LDP on MPLS interface ve 100, enter commands such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface ve 100
NetIron(config-mpls)# ldp-enable
```

Syntax: [no] ldp-enable

The **no** option removes LDP on an MPLS interface, including LDP on an MPLS VE interface.

Setting the LDP hello interval on an MPLS VE interface (link Only)

NOTE
For more information on setting the LDP hello interval on physical interfaces, refer to "Setting the LDP Hello Interval per-Interface (link Only)" on page 1418.

You can set the LDP Hello Interval on an MPLS enabled VE interface. This option is only available for Link LDP sessions. The following example configures LDP hello-interval to 30 seconds for MPLS interface ve 100.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface ve 100
NetIron(config-mpls-if-ve-100)# ldp-params
NetIron(config-mpls-if-ve-100-ldp-params)# hello-interval 30
```

Syntax: [no] hello-interval <*seconds*>

The <*seconds*> variable specifies the value in seconds of the Hello Interval that you are configuring on an MPLS VE interface for LDP Link Hello messages. The LDP hello interval can be from 1 - 32767 seconds. The default value for LDP hello interval is 5 seconds.

The **no** option removes a previously configured LDP Hello Interval.

Setting the LDP hello holdtime on an MPLS VE interface (link only)

NOTE
For more information on setting the LDP hello holdtime on physical interfaces, refer to "Setting the LDP Hello Holdtime per-interface (link only)" on page 1420.

You can set the LDP Hello Holdtime on an MPLS enabled VE interface. This holdtime value is sent in Hello messages from the interface. This option is available for Link LDP sessions only. The following example configures LDP hello-timeout to 18 seconds for MPLS interface ve 100.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface ve 100
NetIron(config-mpls-if-ve-100)# ldp-params
NetIron(config-mpls-if-ve-100-ldp-params)# hello-timeout 18
```

Syntax: [no] hello-timeout <*seconds*>

The value configured in the <*seconds*> variable is the LDP Hello Timeout value that will be sent in LDP Hello messages from this interface. The range for this value is 1 – 65535 seconds. The default value is 15 seconds.

The **no** option removes a previously configured LDP Hello Timeout value and sets the value as described in "Determining the LDP Hold Time on an MPLS interface" on page 1419.

Bandwidth computation for an MPLS VE interface

The maximum reservable bandwidth for a VE interface is computed based on minimum speed of all active members on a physical port. If one of the member ports is a trunk port, MPLS computes the trunk bandwidth before computing the VE bandwidth. The bandwidth of a trunk port is the sum of all active physical member ports of the trunk. For example, there are two ports (One port is 10 gig and other port is 1 gig), and one trunk port configured on a VE interface. The trunk is carrying two ports, and each port is 1 gig. To calculate the bandwidth of the trunk, you take the sum of all active ports on a physical port. In this example, the bandwidth of the trunk is equal to 2 gig. To calculate the bandwidth of the VE interface, take the minimum of all active port members. In this example, the bandwidth of the VE interface is 1 gig.

Configuration Considerations

A VE interface bandwidth must be recomputed when any one of the following occurs:

- A new member port is added to a VLAN associated with a VE interface.
- A new member port is removed from a VLAN associated with a VE interface.
- When a member port of a VLAN associated with a VE interface is up.
- When an active member port of a VLAN associated with a VE interface that is down.

A physical port can be part of more than one VE interface. Each VE interface assumes that it has a full amount of reservable bandwidth for a physical port. However, the amount of reservable bandwidth on one VE will not be reflected on another VE interface even though both VE interfaces share the same physical port. For example, if there are two VE interfaces; VE1 and VE2. Each VE interface supports the same amount of reservable bandwidth of 1Gbps. The amount of reservable bandwidth used to set up LSPs on VE1 is not reflected in the amount reservable bandwidth that is available on VE2. This will result in an excess amount of reservable bandwidth that can be supported on a physical port. This will cause data traffic to be dropped.

The default bandwidth for a VE interface is computed automatically, and is based on the underlying physical links. You can now override the default bandwidth for a VE interface by executing the **reservable-bandwidth** command. The following example demonstrates how to override the default behavior by configuring reservable bandwidth to 400mbps on MPLS interface ve 100.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface ve 100
NetIron(config-mpls-if-ve-100)# reservable-bandwidth 400000
```

Syntax: [no] reservable-bandwidth <*number*>

The <*number*> variable refers to the amount of reservable bandwidth that is supported on an MPLS interface, including an MPLS enabled VE interface. The range for this value is 0 - 80000000 kbps. The **no** option removes all configuration for reservable bandwidth on an MPLS enabled VE interface.

RSVP message authentication on an MPLS VE interface

Support was added for RSVP message authentication using MD5 as described in RFC 2747. It is implemented on the PowerConnect routers to prevent spoofing of RSVP messages. All inbound RSVP messages on an interface must contain RSVP Integrity object for getting authenticated and accepted by RSVP. Inbound RSVP messages with no Integrity object, or an **incorrect** integrity object will be dropped by RSVP. All outbound RSVP messages on an interface contain an RSVP Integrity object. For more information on RSVP message authentication, refer to "RSVP message authentication" on page 1327.

Configuring RSVP message authentication on an MPLS VE interface

NOTE
For more information on configuring RSVP message authentication on physical interfaces, refer to **"Configuring RSVP message authentication"** on page 1328.

RSVP Message Authentication is disabled by default. This authentication method uses MD5 for an MPLS VE interface. The following example configures RSVP message authentication for MPLS interface ve 100.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface ve 100
NetIron(config-mpls-if-ve-100)# rsvp-authentication key private
```

Syntax: [no] **rsvp-authentication key** <*string*>

The <*string*> variable specifies a text string of up to 64 characters that is encrypted and used for RSVP message authentication.

Specifying a bypass LSP for an MPLS VE interface

You can create a bypass LSP by using the **bypass-lsp** command. This is used for facility backup FRR. In the context of bypass LSP, you can configure an MPLS interface as an exclude (protected) interface against resource failures using a bypass LSP. You can specify a VE interface as exclude-interface. If a protected LSP egress interface is a VE interface, then any fault on a VE interface could trigger FastReroute. The following example configures protection for MPLS interface ve 100 using facility backup FRR.

```
NetIron(config)# router mpls
NetIron(config-mpls)# bypass-lsp to4
NetIron(config-mpls-bypasslsp-to4)# exclude-interface ve 10
```

Syntax: [**no**] **exclude-interface ethernet** <*slot/port*> [**ethernet** <*slot/port*> | **to** <*slot/port*>] | **pos** <*slot/port*> [**pos** <*slot/port*> | **to** <*slot/port*>] | **ve** <*vid*>]

By default, a VE interface is not protected. The **ve** parameter allows you to configure a ve interface as exclude-interface specified by <*vid*>.

Setting up signalled LSPs

An LSP consists of an actual path of MPLS routers through a network, as well as the characteristics of the path, including bandwidth allocations and routing metrics. There are two kinds of LSPs: signalled and static. Signalled LSPs are configured at the ingress LER. When you enable a signalled LSP, RSVP causes resources to be allocated on the other routers in the LSP.

Configuring a signalled LSP consists of the following tasks:

- Specifying a path for the LSP to follow (optional)
- Setting parameters for the signalled LSP
- Specifying which packets are to be forwarded along the LSP (optional)

Setting up paths

A ***path*** is a list of router hops that specifies a route across an MPLS domain. Once you create a path, you can create signalled LSPs that see the path. Paths are configured separately from LSPs so that a path may be specified once and then used by several LSPs that see the path by name. An LSP may specify a primary and one or more redundant paths.

A path is always configured at the ingress LER and assumes that the ingress LER is the beginning of the path. A path can contain any number of ***nodes***, which correspond to MPLS-enabled routers in the network. Each node has one attribute: whether it is ***strict*** or ***loose***. A strict node means that the router must be directly connected to the preceding node. A loose node means that there can be other routers in between.

Creating a path is not absolutely necessary when configuring an LSP. If you configure a signalled LSP without naming a path, CSPF uses only information in the Traffic Engineering Database (TED), as well as the user-configured attributes and requirements of the LSP to calculate the path. Refer to "How CSPF calculates a traffic-engineered path" on page 1301 for more information. If the LSP has been configured not to use CSPF, the path between the ingress and egress LERs is determined using standard hop-by-hop routing methods, as if the path consisted of a single loose node.

The following commands set up a path called sf_to_sj that has four nodes.

```
NetIron(config-mpls)# path sf_to_sj
NetIron(config-mpls-path)# strict 216.150.1.1
NetIron(config-mpls-path)# strict 216.150.1.2
NetIron(config-mpls-path)# loose 64.1.1.1
NetIron(config-mpls-path)# strict 64.100.1.1
NetIron(config-mpls-path)# exit
```

Syntax: [no] path <*path name*>

Syntax: [no] strict | loose <*ip address*>

The path is assumed to start from the local node. You specify the nodes in order from ingress to egress. Specifying the local node itself as the first node in the path is optional. Further, the final node does not necessarily have to be the egress LER in the LSP. (The egress LER is specified at the LSP configuration level with the **to** command.) If the final node in the path differs from the egress LER, the hop between the final node in the path and the egress LER is treated as a hop to a loose node; that is, standard IP routing is used to determine the path between the final node and the egress LER.

The IP address defines an LSR and can be any interface address or a loopback interface address on the LSR.

The **strict** and **loose** parameters are relative to the preceding node. In the sf_to_sj path defined above, LSR 216.150.1.2 is a strict node; it must be directly connected to LSR 216.150.1.1. LSR 64.1.1.1 is a loose node; this means there can be other routers between LSR 216.150.1.2 and 64.1.1.1. When specifying a strict node, you should make sure that the LSR is actually directly connected to the preceding node.

Modifying a path

Once you have created a path, you can insert or delete nodes from it. For example, to delete a node from the sf_to_sj path defined above.

```
NetIron(config-mpls)# path sf_to_sj
NetIron(config-mpls-path)# delete loose 64.1.1.1
NetIron(config-mpls-path)# exit
```

Syntax: [no] delete strict | loose <*ip address*>

To insert a node into a path.

```
NetIron(config-mpls)# path sf_to_sj
NetIron(config-mpls-path)# insert strict 216.150.1.1 before 216.150.1.2
NetIron(config-mpls-path)# exit
```

Syntax: [no] insert strict | loose <*ip address*> **before** <*ip address*>

The **insert** command allows a new node to be inserted in front of an existing node within the path. In this example, the **insert strict 216.150.1.1 before 216.150.1.2** command assumes that 216.150.1.2 is already in the path and inserts 216.150.1.1 before it.

NOTE
When you modify a path, the changes are not carried over to active LSPs that see the path until the LSPs are deactivated and reactivated. For example, path sj_to_sf may be used by an LSP called lsp1. After lsp1 has been activated, any changes to path sj_to_sf do not cause the route followed by lsp1 to be modified. To get the LSP to use the modified path, you must deactivate and then reactivate lsp1.

Deleting a path

To delete an entire path from the LSR's configuration, enter a command such as the following.

```
NetIron(config-mpls)# no path sf_to_sj
```

Syntax: [no] path <*path name*>

Configuring signalled LSP parameters

Once you have configured a path, you can configure signalled LSPs that see it. An LSP's configuration can specify not only the path that label-switched packets follow in a network, but also the characteristics of the path, the resources allocated along the path, and actions applied to the packets by the ingress or egress LERs.

You can perform the following tasks when configuring a signalled LSP:

- Performing a Commit for an LSP
- Creating the LSP
- Specifying an egress LER for the LSP

- Specifying a primary path for the LSP (optional)
- Configuring secondary or hot-standby paths for the LSP (optional)
- Setting aliases for the egress LER (optional)
- Setting a Class of Service (CoS) value for the LSP (optional)
- Allocating bandwidth to the LSP (optional)
- Configuring the setup and hold priority for the LSP (optional)
- Setting a metric for the LSP (optional)
- Including or excluding administrative groups from LSP calculations (optional)
- Limiting the number of hops the LSP can traverse (optional)
- Specifying a tie-breaker for selecting CSPF equal-cost paths (optional)
- Disabling the Record-Route function (optional)
- Disabling CSPF path calculations (optional)
- Configure Maximum Packet Size without Fragmentation
- Enabling the LSP
- Disabling the LSP
- Generating Traps and Syslogs for LSPs

Performing a commit for an LSP configuration command

For LSP configuration commands to take effect, either an explicit or implicit commit must be performed. These are performed as shown in the following:

Performing and explicit commit

You can perform an explicit commit within the configuration of a specified LSP using the **commit** command. The following example demonstrates the creation of an LSP named “samplelsp” and its primary and secondary paths. After the configuration is entered, the commit command is executed to activate the configuration.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp samplelsp
NetIron(config-mpls-lsp-samplelsp)# primary-path pathprimary
NetIron(config-mpls-lsp-samplelsp)# secondary-path pathsecondarya
NetIron(config-mpls-lsp-samplelsp)# secondary-path pathsecondaryb
NetIron(config-mpls-lsp-samplelsp)# select manual pathsecondaryb
NetIron(config-mpls-lsp-samplelsp)# commit
```

Syntax: [no] commit

Performing and implicit commit

The **reoptimize** and **reoptimize_timer** commands allow you to perform an implicit commit.

Using the **reoptimize** command, you can activate all pending LSP configuration changes for specified LSP or use the **all** option to activate all pending LSP configuration changes for all of the LSPs configured on the router. Configuration of this command is described in “Reoptimizing LSPs” on page 1349.

The **reoptimize_timer** command allows you to set a periodic timer to perform the reoptimize function for all LSPs at an interval specified in seconds. Configuration of this command is described in “Time-triggered reoptimizing” on page 1349.

Creating an LSP

To create a signalled LSP and enter the LSP configuration level, enter commands such as the following.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)#
```

Syntax: [no] lsp <*name*>

Specifying the egress LER

Each LSP requires one and only one egress LER. The egress LER is the router from which packets exit the MPLS domain in this LSP. After the LSP is successfully established, the address of the egress LER is installed as an internal host route on the ingress LER, allowing the ingress LER to direct BGP next-hop traffic into the LSP. The destination address does not necessarily have to be the final node in the primary path specified for the LSP. If the final node in the path differs from the destination address, the hop between the final node in the path and the egress LER is treated as a loose hop.

To specify 64.100.1.1 as the address of the egress LER for LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# to 64.100.1.1
```

Syntax: to <*ip address*>

The egress LER is the only required parameter in an LSP. All other parameters are optional.

NOTE
If OSPF is used as the IGP, the egress LER should advertise the tunnel destination in type 1 (router) LSA in order for the LSP to be properly mapped by CSPF. To ensure that this happens, connect to the egress LER and enable OSPF on the interface which has the IP address of the tunnel destination. (See "Assigning interfaces to an area" on page 874 for details.) If none of the interfaces on the egress LER has the IP address of the tunnel destination (e.g., if the tunnel destination address is the egress LER's router ID rather than an interface address -- to manually set the router ID, see "Changing the router ID" on page 684), then the tunnel destination address must be included in the router address TLV in the type 10 LSA originated by the egress LER. This is accomplished by setting the egress LER's traffic engineering policy to OSPF with the **traffic-engineering ospf** command (see "Enabling OSPF-TE LSAs for MPLS interfaces" on page 1321).

NOTE
If IS-IS is used as the IGP, the egress LER should advertise the tunnel destination in Extended IP Reachability TLV 135 in order for the LSP to be properly mapped by CSPF. To ensure that this happens, connect to the egress LER and enable ISIS on the interface which has the IP address of the tunnel destination. (See "Disabling and enabling IS-IS on an interface" on page 968 for details.) If none of the interfaces on the egress LER has the IP address of the tunnel destination (e.g., if the tunnel destination address is the egress LER's router ID rather than an interface address -- to manually set the router ID, see "Changing the router ID" on page 684), then the tunnel destination address must be included in Traffic Engineering router ID TLV 134 in the LSP originated by the egress LER. This is accomplished by setting the egress LER's traffic engineering policy to IS-IS with the **traffic-engineering isis level** command (see "Enabling IS-IS LSPs with TE extensions for MPLS interfaces" on page 1322).

Specifying a source address for an LSP

You can optionally specify a source IP address for a signalled LSP. RSVP path messages carry this address.

To specify a source IP address of 1.2.3.4 for LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# from 1.2.3.4
```

Syntax: **from** <*ip address*>

The **from** command specifies the source IP address to be carried in RSVP Path messages for the LSP. This command is optional. If the **from** command is specified, then the address is always carried in RSVP Path messages as the source IP address for the LSP. If the **from** command is not specified, then when the LSP is enabled, the device dynamically determines the source address of the LSP (using the device's router ID or the address of the first loopback as the source address).

Note that the IP address specified in the **from** command affects only the address carried in the RSVP Path messages for the LSP. It does not affect the outgoing interface (and thus the actual path) that the Path messages are sent out.

Specifying the primary path for an LSP

The primary path is the route that packets generally travel when going through an LSP. You can specify a user-defined path or no path at all. Refer to "Setting up paths" on page 1333 for information on defining a path. Once the LSP is enabled, the ingress LER attempts to signal the other LSRs in the path so that resources can be allocated to the LSP. If you do not specify a primary path, the path used in the LSP is the shortest path to the egress LER, as determined from standard IP routing methods, or CSPF if it is enabled.

To specify the sf_to_sj path as the primary path for LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# primary-path sf_to_sj
```

Syntax: [no] **primary-path** <*path name*>

Configuring redundant paths for an LSP

NOTE
This section describes the behavior of redundant paths. However, you can exercise further control over the path selection process by specifying the path selection mode and preferred path using the **select-path** command. This process is described in detail in "Configuring path selection" on page 1339.

A signalled LSP has a primary path, which is either user-defined or computed by the ingress LER. Optionally, you can configure one or more redundant paths to serve as a backup. If the primary path fails, traffic for the LSP can be forwarded over the redundant path. When no redundant path is configured for the LSP, if the primary path fails, the ingress LER automatically attempts to compute a new path to the egress LER, establish the new path, and then redirect traffic from the failed path to the new path.

Configuring a redundant path allows you to exercise greater control over the rerouting process than if the ingress LER simply calculated a new path to the egress LER. When a redundant path is configured, if the primary path fails, the ingress LER attempts to establish the redundant path. As with the primary path, a redundant path follows an explicit route of loose or strict hops.

By default, the redundant path is established only when the primary path fails. You can optionally configure a redundant path to operate in ***hot-standby*** mode. A hot-standby path is established at the same time the primary path in the LSP is established. Resources are allocated to the hot-standby path, although no packets for the LSP are sent over the hot-standby path until the primary path fails. When the primary path fails, the already-established hot-standby path immediately takes over from the primary path. Since the hot-standby path is already active, service outages that can arise from the process of signaling and establishing a new path are eliminated.

After the redundant path has been activated, the ingress LER continues to try to connect to the egress LER over the primary path, either indefinitely or up to the configured retry limit. If a connection over the primary path can be established, the redundant path is deactivated, and traffic for the LSP is again sent over the primary path. Once the primary LSP becomes available again, the redundant path is torn down; if the path is a hot-standby path, it reverts to its backup status.

You can configure multiple redundant paths. When the primary path fails, the ingress LER attempts to establish a connection to the egress LER using the first redundant path configured for the LSP. If a connection cannot be established using the first redundant path, the second redundant path is tried, and so on. If a connection cannot be established after trying each redundant path in the configuration, the first redundant path is tried again, and the process repeats. (This behavior can be further modified using the **select-path** command; see "Configuring path selection" on page 1339.)

To configure a secondary path, first create a path, as described in "Setting up paths" on page 1333. After you create the path, you can specify that it is to be used as a redundant path. For example, the following commands cause a path called alt_sf_to_sj to be used if the primary path in LSP tunnel1 fails.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# secondary-path alt_sf_to_sj
NetIron(config-mpls-lsp-sec-path)#
```

Syntax: [no] secondary-path *<path name>*

Issuing the **secondary-path** command enters the secondary path configuration level. From this level, you can specify that this path is to operate in hot standby mode.

Example

```
NetIron(config-mpls-lsp-sec-path)# standby
```

Syntax: [no] standby

Once the LSP is enabled, both the primary and hot-standby paths are activated, although packets are directed over only the primary path.

NOTE
At the secondary path level, you can configure separate values for the following parameters: Class of Service (CoS), setup and hold priority, bandwidth allocations, and inclusion or exclusion of interfaces in administrative groups. If you do not configure these parameters at the secondary path level, the secondary path will use the default values for these parameters.

Configuring path selection

You can exercise further control over the paths used by an LSP by setting the select mode and by specifying a preferred path using the **select-path** command as described below.

By default, an LSP with primary and secondary paths configured immediately uses the primary path. If the primary path fails, a secondary (redundant) path is used. If the primary path comes back up, traffic reverts to the primary path and the secondary (redundant) path returns to a back-up state. However, path selection can be configured to operate in any of the following three modes:

- **auto select mode** – This is the default mode of the router and no special configuration is required. When this mode is operating, the router will always try to use the primary path to carry traffic when the primary path has stayed operating in the working state for at least the amount of time specified in **revert-timer** configuration command. If no **revert-timer** is configured for the LSP, a value of zero second is used which causes immediate switching of the path.
- **manual select mode** – In this mode, traffic is switched to a user-specified path after the selected path has stayed operating in the working state for at least the amount of time specified in **revert-timer** configuration. In **manual select** mode, traffic stays on the selected path as long as the path remains in working condition and only switches to an alternative path, such as the primary path, when the selected path experiences a failure. Once the selected path comes back into working condition for the amount of time specified by the revert-timer configuration, traffic is switched back to it.

  When an LSP is configured in manual select path mode with at least one other hot standby secondary path, the operation is as follows: if the selected path goes down, the system will try to bring up one hot standby secondary path to protect the primary path, but if selected path is up, system will bring down the hot standby secondary path since the selected path is already serving as a hot standby for the primary path.
- **unconditional select mode –** In this mode, traffic is switched to and stays on the selected path regardless of the path's condition even if it is in a failure state. The main difference between manual and unconditional select mode is the test of the working condition of the user selected path. When configured in unconditional mode, the router starts the signaling for the selected path if has not already done so and brings down all other paths; this includes the primary path and the path carrying traffic if it is not the selected path. Because the speed at which the selected path comes up cannot be guaranteed, traffic forwarding might be disrupted.

NOTE
The **auto-select** and **manual-select** mode configurations use the **revert-timer** configuration that is described in "Configuring a Path Selection Revert Timer" on page 1340.

The following example configured the LSP named "samplelsp" with a primary path named "pathprimary" and two secondary paths named "pathsecondarya" and pathsecondaryb". The path named "pathsecondaryb" is configured as a selected path in the "manual select" mode.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp samplelsp
NetIron(config-mpls-lsp-samplelsp)# primary-path pathprimary
NetIron(config-mpls-lsp-samplelsp)# secondary-path pathsecondarya
NetIron(config-mpls-lsp-samplelsp)# secondary-path pathsecondaryb
NetIron(config-mpls-lsp-samplelsp)# select-path manual pathsecondaryb
NetIron(config-mpls-lsp-samplelsp)# commit
```

After configuring this example, traffic for “samplelsp” travels over the “pathsecondaryb” path whenever this path is in working condition because no revert-timer has been configured. If a revert-timer is configured, the router waits for the “pathsecondaryb” path to be up for at least the amount of time specified in the configuration of the **revert-timer** command. If the select mode is changed to **unconditional**, as shown in the following, traffic will be switched to the “pathsecondaryb” path regardless of its working condition.

```
NetIron(config-mpls-lsp-samplelsp)# select-path unconditional pathsecondaryb
```

Syntax: [**no**] **select-path** { **manual** | **unconditional** } { <*path-name*> | **primary** }

The **no** option returns an LSP to the default auto select mode if it has been previously configured to the manual select mode or unconditional select mode.

The <*path-name*> variable is the name of the path that you want to assign manual select mode or unconditional select mode to. You can optionally specify the primary path by using the **primary** keyword.

The **manual** option configures the specified path to operate in the manual select mode. If you select **primary** as the specified path with the **manual** option, the primary path is selected as the preferred path, which is the same as the default operation.

The **unconditional** option configures the specified path to operate in the unconditional select mode. If you select **primary** as the specified path with the **unconditional** option, the primary path is selected as the preferred path regardless of the condition of the primary path.

Configuration changes made to the select mode do not take effect for an already enabled LSP until the change is activated implicitly using the **commit** command or explicitly using a **reoptimize** command as described in “Performing a commit for an LSP configuration command” on page 1335 or a system reboot is performed.

NOTE
When you configure a primary path to be the selected path, a message is generated that states that it is already the default system behavior because the primary path is the default preferred path. In this instance, no configuration information is saved in the configuration file.

Configuring a Path Selection Revert Timer

The Path Selection Revert Timer feature provides an option to stabilize a path before traffic is switched to it. Without a configured Path Selection Revert Timer, the router switches between a primary and secondary path immediately after the current working path goes down. A problem with this mode of operation is that it can cause flapping if the current path goes up and down frequently. Also, the LSP to which the route is switching traffic might be unstable, which causes the router to fail back to the current LSP almost immediately.

The Path Revert Timer insures the stability of the LSP to which the traffic is switched by specifying the number of seconds that the LSP must be running before it actually carries traffic.

To configure a Path Selection Revert Timer for an LSP, use the **revert-timer** command in the LSP configuration context, as shown in the following.

```
NetIron(config-mpls)# lsp samplelsp
NetIron(config-mpls-lsp-samplelsp)# revert-timer 10
```

Syntax: [**no**] **revert-time** <*timer-value*>

The <*timer-value*> value is the number of seconds that the router waits after the primary or selected path comes up before traffic reverts to that path. The range is 1–65,535 seconds.

Usage considerations:

- The **revert-time** command has no effect on the unconditional select mode. Traffic is unconditionally switched to the user-selected path and stays on it.
- The path stability test used with the Revert Timer feature is based on the uptime of the latest instance of the path. This value can be different when the selected path has gone through a "make-before-break" procedure.
- For an LSP going through re-optimization, the new LSP does not carry traffic until the revert timer expires.
- When a user changes the revert timer, the basis of counting is the uptime of the path and is independent of the sequence or combination of configurations. Take, for example, a path that is configured in the manual select mode to be a secondary path with a revert-timer of 10 seconds. After the secondary path comes up, a 10-second timer starts, but after 5 seconds, the user changes the revert-timer value to 4. Now the path has already been stable beyond the new configured revert-timer, so the original timer is canceled and traffic immediately switches over. However, if the user were to change the revert-timer value to 8 seconds after running for 5 seconds, the existing count would terminate and start a new count of 3 seconds from the moment the first count terminated.

To configure a Path Selection Revert Timer, for an LSP, use the **revert-timer** command within the LSP configuration as shown in the following.

```
NetIron(config-mpls)# lsp samplelsp
NetIron(config-mpls-lsp-samplelsp)# revert-timer 10
```

Syntax: [**no**] **revert-time** <*timer-value*>

The <*timer-value*> value specifies an amount of time in seconds that the router will wait after the primary or selected path comes back up before reverting to it.

Setting a Class of Service value for the LSP

The 3-bit EXP field in the MPLS header can be used to define a Class of Service (CoS) value for packets travelling through the LSP. You can manually set a CoS value for the LSP. The CoS value that you set is applied to the CoS (EXP) field in the MPLS header of all packets entering this LSP. This lets all packets travelling through an LSP to be treated with the same priority as they travel the MPLS domain. You can assign the LSP a CoS in the range 0–7.

To assign a CoS value of 7 (highest priority) to all packets traveling through LSP tunnel 1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# cos 7
```

Syntax: [**no**] **cos** <*number*>

The MPLS CoS value is used for determining priority within an MPLS domain only, so when the label is popped, the CoS value in the MPLS header is discarded; it is not copied back to the IP ToS field.

Allocating bandwidth to an LSP

You can specify the allocation of bandwidth for an LSP, including the maximum and average rates for packets that travel over it. Allocating bandwidth to an LSP lets the LSRs determine how much bandwidth the LSP can consume and how much of the available bandwidth resources can be advertised by using OSPF-TE LSAs.

You can specify an average <*mean-rate*> kbps for the data on the LSP. When necessary, data can travel at <*max-rate*> kbps, as long as the burst sent at the maximum rate contains no more than <*max-burst*> bytes.

To set the maximum rate of packets that can go through an LSP (in Kbits/sec).

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# traffic-eng max-rate 20
```

Syntax: [**no**] **traffic-eng max-rate** <*rate*>

To set the average rate of packets that can go through an LSP (in Kbits/sec).

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# traffic-eng mean-rate 10
```

Syntax: [**no**] **traffic-eng mean-rate** <*rate*>

To set the maximum size (in bytes) of the largest burst the LSP can send at the maximum rate.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# traffic-eng max-burst 10
```

Syntax: [**no**] **traffic-eng max-burst** <*bytes*>

Configuring a peiority for a signalled LSP

You can specify a priority for each signalled LSP for which this is the ingress LER. The priority determines the relative importance of the LSP during setup or preemption. The priority for an LSP has two components the setup priority and the hold priority.

When multiple LSPs are enabled at the same time, such as when the device is booted, LSPs that have a higher setup priority are enabled before LSPs that have a lower setup priority.

If an LSP is assigned a high setup priority, it may preempt an LSP that is already established, causing resources assigned to the lower priority LSP to be diverted to the higher priority LSP. The hold priority specifies how likely an established LSP is to give up its resources to another LSP. To be preempted, an LSP must have a lower hold priority than the preempting LSP's setup priority. In addition, an established LSP can be preempted by a higher priority LSP only if it would allow the higher priority LSP to be established successfully.

To configure LSP tunnel1 with a setup priority of 6 and hold priority of 1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# priority 6 1
```

Syntax: **[no] priority** <*setup-priority*> <*hold-priority*>

Possible values are 0 (highest priority) through 7 (lowest priority). An LSP setup priority must be lower than or equal to the hold priority. The default LSP setup priority is 7, and the hold priority is 0.

Assigning a metric to the LSP

You can assign a metric to the LSP, which can be used by routing protocols to determine the relative preference among several LSPs towards a given destination.

To assign a metric of 5 to LSP tunnel1

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# metric 5
```

Syntax: **[no] metric** <*number*>

The metric has a range of 1–65535. By default, all LSPs have a metric of 1. A lower metric is preferred over a higher one. If multiple LSPs have the same destination LSR, and they have the same metric, the traffic load is shared among them.

Including or excluding administrative groups from LSP calculations

Administrative groups, also known as resource classes or link colors, let you assign MPLS-enabled interfaces to various classes. When a device uses CSPF to calculate the path for an LSP, it takes into account the administrative group to which an interface belongs; you can specify which administrative groups the device can include or exclude for this calculation.

For example, to include interfaces in either of the administrative groups “gold” and “silver” in the path calculations for LSP tunnel1, do the following.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# include-any gold silver
```

Syntax: [no] include-any <*groups*>

The value specified for <*groups*> can be one or more valid administrative group names or numbers. In this example, the device includes any of the interfaces that are members of groups “gold” or “silver” when calculating the path for this LSP. Only those interfaces in the “gold” or “silver” groups are considered for the LSP. Interfaces that are not part of these groups, as well as interfaces that are not part of any group, are eliminated from consideration.

To exclude interfaces in either administrative group “gold” or “silver” when the path for LSP tunnel1 is calculated.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# exclude-any gold silver
```

Syntax: [no] exclude-any <*groups*>

In this example, the device excludes any interface that is a member of group “gold” or “silver” when it calculates the path for this LSP. Only interfaces that are not part of either group are considered for the LSP.

To specify that an interface must be a member of both the “gold” or “silver” administrative groups in order to be included in the path calculations for LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# include-all gold silver
```

Syntax: [no] include-all <*groups*>

In this example, an interface must be a member of all the groups specified in the **include-all** command in order to be considered for the LSP. Any interface that is not a member of all the groups is eliminated from consideration.

Limiting the number of hops the LSP can traverse

By default, the path calculated by CSPF can consist of no more than 255 hops, including the ingress and egress LERs. You can optionally change this maximum to a lower number.

For example, to limit CSPF to choosing a path consisting of no more than 20 hops for LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# hop-limit 20
```

Syntax: [no] hop-limit <*number*>

The range for the number of hops is 0–255.

Specifying a tie-breaker for selecting CSPF equal-cost paths

CSPF may calculate multiple, equal-cost paths to the egress LER. When this happens, the device chooses the path whose final node is the physical address of the destination interface. If more than one path fits this description, by default, the device chooses the path with the fewest hops. If multiple paths have this number of hops, the device chooses one of these paths at random. You can optionally configure the device to choose the path that has either the highest available bandwidth or the lowest available bandwidth.

For example, the following commands cause CSPF to select the path with the highest available bandwidth when choosing among equal-cost paths calculated for LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# tie-breaking least-fill
```

Syntax: [no] tie-breaking least-fill | most-fill | random

The **least-fill** parameter causes CSPF to choose the path with the highest available bandwidth (that is, the path with the least utilized links).

The **most-fill** parameter causes CSPF to choose the path with the lowest available bandwidth (that is, the path with the most utilized links).

The **random** parameter causes CSPF to choose the path randomly from the equal-cost paths. This is the default.

Disabling the record route function

The RSVP RECORD_ROUTE object (RRO) allows an LSP's path to be recorded. An RRO consists of a series of subobjects that can contain the addresses of the LSRs in the path. This information can be viewed with the **show mpls lsp detail** command. The path information is recorded in the RRO by default, but you can disable path recording.

To disable path recording in the RRO.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# no record
```

Syntax: [no] record

Disabling CSPF path calculations

By default, CSPF is enabled for signalled LSP calculations. That is, if the device receives OSPF-TE LSAs, it places the traffic engineering information from them in its Traffic Engineering Database (TED). When the device is the ingress LER for the LSP, it uses the information in the TED to help determine a path for the LSP. If all nodes in your network are not capable of sending out OSPF-TE LSAs, you may want to disable CSPF for the LSP.

To disable constraint-based path selection for LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# no cspf
```

Syntax: [no] cspf

Configuring the maximum packet size

This feature allows you to set a maximum IP packet size for packets that traverse an LSP without being fragmented. It can be configured for both primary and secondary paths.

To configure a maximum IP packet size for an LSP, enter commands such as the following.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# ipmtu 1500
```

Syntax: [no] ipmtu <*packet-size*>

The <*packet-size*> variable specifies the maximum packet size in bytes for IP packets transiting the LSP without being fragmented.

Enabling a signalled LSP

After you set the parameters for the signalled LSP, you can enable it. Enabling the LSP causes the path to be set up and resources reserved on the LSRs in the LSP's primary path. Enabling the LSP is the final step in configuring it.

To enable LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# enable
```

Syntax: [no] enable

Disabling an LSP

Disabling an LSP de-activates it, but does not remove the LSP from the device's configuration. (To remove the LSP from the device's configuration, use the **no lsp** <*name*> command.) To make changes to an active LSP, first disable the LSP, modify parameters on the LSP, and then enable the LSP.

To disable LSP tunnel1.

```
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# disable
```

Syntax: [no] disable

Resetting LSPs

The **clear mpls lsp** command allows you to reset an RSVP LSP session. Changes in the routing table after an LSP path is established do not take effect unless the LSP is brought down and then brought up again. After you reset the LSP, it will realign to the new routing topology. The **clear mpls lsp** command can be used on the ingress LSR of the LSP.

Resetting normal LSPs

The **clear mpls lsp** command allows you to reset normal LSPs. You have the option of supplying the **primary**/**secondary** parameter for a normal LSP to reset only the primary/secondary path of the LSP.

To reset/clear a bypass RSVP LSP session.

```
NetIron(config-mpls)# clear mpls bypass-lsp <bypass-lsp-name>
```

When you reset an LSP with the `clear mpls lsp` command, the following information message is displayed.

`"Disconnecting signaled LSP <name>"`

"Connecting signaled LSP <*name*>"

Syntax: clear mpls lsp <*lsp-name*> **[primary | secondary]**

If a **primary** or **secondary** optional keyword is not specified when you reset a normal LSP, then both the **primary** and **secondary** LSP paths associated with the <*lsp-name*> will be reset and restarted

Resetting Bypass LSPs

This command allows you to reset bypass LSPs.

NOTE
The **primary** or **secondary** optional keywords are not applicable for bypass LSPs.

Reset LSP considerations

The **clear mpls lsp** and **clear mpls bypass-lsp** commands reset and restart an MPLS RSVP LSP.

NOTE
These commands are disruptive. Data traffic forwarding is impacted as the LSP is not in active state for sub-seconds after teardown. Resetting an LSP could trigger a series of actions depending upon the current state of the LSP.

The following describes the actions and state changes when an LSP is reset.

Resetting an LSP will also reset the associated backup/detour LSPs:

- Resetting the primary path of an LSP will cause the secondary LSP path to become active, if a hot-standby secondary path for the LSP is available. However, if the primary path comes up after the reset operation, the active path will switchover from the secondary to the primary again. If the "revert-timer" is configured, the LSP path switchover may be dampened and will obey the usual revert-timer rule. There is no change in the revert-timer behavior due to the reset LSP feature.

NOTE
The above state changes are described here for informational purposes only. There could be several other intermediate state changes that are not listed here.

- Resetting the primary path of an adaptive LSP will also reset the "other" new instances of the LSP's primary path, if available at the time of reset.
- Resetting the secondary path for an LSP will reset the current secondary path of the LSP. It will also reset the selected secondary path, if available at the time of reset.
- Resetting the secondary path for an LSP whose primary path is down may trigger the secondary path selection process to choose a new secondary path. If a new secondary path is found, it will be signaled and may become the active path. If no secondary paths are found, then the current secondary may become the active path again after successful RSVP signaling.
- The primary path is UP but not active, and the secondary path is UP and active. The secondary to primary switchover occured because the revert-timer has been configured (using a large value). Resetting the secondary LSP path will still force the path switchover from secondary to primary path in spite of the revert-timer configuration.

- For an adaptive LSP, if reset is performed before the **commit** command, then the LSP will be reset and will come-up with a new set of configuration parameters. However, this will be disruptive for data traffic unlike the **commit** command, because the current instance of the LSP will be reset while there is no new instance of the LSP available (because the **commit** command has not been executed yet)

Generating traps and syslogs for LSPs

Multi-Service IronWare software supports the ability to enable and disable SNMP traps and syslogs for LSPs. LSP traps and syslogs are enabled by default.

To enable LSP traps after they have been disabled, enter the following command.

```
NetIron(config)# snmp-server enable traps mpls lsp
```

Syntax: [no] snmp-server enable traps mpls lsp

Use the **no** form of the command to disable LSP traps.

To enable LSP syslogs after they have been disabled, enter the following command.

```
NetIron(config)# log enable mpls lsp
```

Syntax: [no] log enable mpls lsp

Use the **no** form of the command to disable the syslog for LSPs.

Configuring an adaptive LSP

The Multi-Service IronWare software supports Adaptive LSPs. Using this feature, you can change the following parameters of an LSP while it is in the enabled state:

- cspf
- exclude-any
- hop-limit
- include-all
- include-any
- primary-path
- priority
- tie-breaking
- traffic-eng

When one of these parameters is changed on a Adaptive LSP, a new instance of the same LSP is signaled using the newly defined parameters. Once the new LSP comes up, traffic is moved to the new LSP instance and the old LSP instance is torn down.

To configure an LSP named to20 as an Adaptive LSP, use the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp to20
NetIron(config-mpls-lsp-to20)# adaptive
```

Syntax: [no] adaptive

Once an LSP is configured to be adaptive, it can have the parameters described above changed. In the following example, the Setup and hold priorities for adaptive LSP to20 are changed to 7 and 1.

```
NetIron(config-mpls)# lsp to20
NetIron(config-mpls-lsp-to20)# priority 7 1
```

The new parameters are not changed for the adaptive LSP until the **commit** command is issued for the LSP.

NOTE
Once the **commit** command has been issued, there may be a 30 ms traffic disruption.

In the following example of the **show mpls lsp** command for lsp to20, the priorities are not changed in the output.

```
NetIron(config-mpls-lsp-to212)#show mpls lsp to212
LSP to212, to 10.5.1.1
 From: 10.4.1.1, admin: UP, status: UP, tunnel interface: tnl1
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0  Adaptive
  Maximum retries: 0, no. of retries: 0
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  OTHER INSTANCE PRIMARY: NEW_INSTANCE admin: DOWN, status: DOWN
  Maximum retries: 0, no. of retries: 0
  Setup priority: 7, hold priority: 1
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl1, outbound interface: e1/2
   Tunnel index: 4, Tunnel instance: 1 outbound label: 3
   Path calculated using constraint-based routing: yes
   Explicit path hop count: 1
    10.2.1.2 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    10.2.1.2
```

The following **commit** command makes the new parameter settings active in Adaptive LSP to20's configuration

```
NetIron(config-mpls)# lsp to20
NetIron(config-mpls-lsp-to20)# commit
```

Syntax: [no] commit

After the commit command runs, you can see that the priorities have changed by using the **show mpls lsp** command for lsp to20.

```
NetIron(config-mpls-lsp-to212)#show mpls lsp to212
LSP to212, to 10.5.1.1
  From: 10.4.1.1, admin: UP, status: UP, tunnel interface: tnl1
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0  Adaptive
  Maximum retries: 0, no. of retries: 0
  Setup priority: 7, hold priority: 1
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl1, outbound interface: e1/2
```

```
Tunnel index: 4, Tunnel instance: 2 outbound label: 3
Path calculated using constraint-based routing: yes
Explicit path hop count: 1
 10.2.1.2 (S)
Recorded routes:
 Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
 10.2.1.2
```

Reoptimizing LSPs

Under ordinary conditions, an LSP path will not change unless the path becomes inoperable. Consequently, the router needs to be directed to consider configuration changes made to an LSP and to optimize the LSP path based on those changes. This is accomplished using the **mpls reoptimize** command as shown in the following.

```
NetIron# mpls reoptimize lsp to20
```

Syntax: [no] mpls reoptimize all | lsp <*lsp-name*>

The **all** option directs the router to reoptimize the paths for all LSPs configured.

The **lsp** option directs the router to reoptimize the path for the LSP specified by the <*lsp-name*>.

NOTE
On reoptimization of an adaptive LSP, LSP accounting statistics might miss the accounting of some of the packets.

Time-triggered reoptimizing

You can set a timer to optimize a specific LSP path on a periodic basis. Upon expiration of this timer, the LSP is optimized for a new path if the new path has a lower cost than the existing path. This timer can be configured when the LSP is in a disabled state, and the timer value can be adaptively changed when the LSP is in an enabled state.

To set the LSP reoptimization timer, use the **reoptimize_timer** command during LSP configuration, as the following shows.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp to20
NetIron(config-mpls-lsp-to20)# reoptimize_timer 1000
```

Syntax: [no] reoptimize_timer <*seconds*>

The <*seconds*> variable specifies the number of seconds from the beginning of one reoptimization attempt to the beginning of the next attempt. The range of values for <*seconds*> is 300–65535.

The **no** option can be used to disable a timer that has been configured. By default, a timer is not configured.

Configuring a reoptimization timer does not interfere with running the manual **reoptimize** command as described in "Reoptimizing LSPs" on page 1349.

NOTE
When upgrading software, configured adaptive LSPs are initialized with no reoptimization timer.

NOTE
This feature does not apply to LSPs within a FRR network.

Configuring MPLS Fast Reroute using one-to-one backup

To configure MPLS Fast Reroute by using the one-to-one backup method for a defined LSP named "frr_tunnel," use the **ffr** command as in the following example.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp frr_tunnel
NetIron(config-mpls-lsp-frr_tunnel)# to 30.1.1.1
NetIron(config-mpls-lsp-frr_tunnel)# primary-path direct_path
NetIron(config-mpls-lsp-frr_tunnel)# secondary-path alt_path
NetIron(config-mpls-lsp-frr_tunnel)# frr
NetIron(config-mpls-lsp-frr_tunnel-frr)# bandwidth 100
NetIron(config-mpls-lsp-frr_tunnel-frr)# hop-limit 20
NetIron(config-mpls-lsp-frr_tunnel-frr)#
```

Syntax: [no] frr

This command enables MPLS Fast Reroute using the one-to-one backup on the LSP under whose configuration it is enabled. Options for this command are described in the sections that follow.

MPLS Fast Reroute using one-to-one backup configuration options

The following options can be set for a MPLS Fast Reroute using one-to-one backup configuration:

- Bandwidth
- Exclude any
- Hop limit
- Include all
- Include any
- Priority

Configuring bandwidth for a MPLS Fast Reroute

To define a bandwidth constraint for the Fast Reroute path, use the following command.

```
NetIron(config-mpls-lsp-frr_tunnel)# frr
NetIron(config-mpls-lsp-frr_tunnel-frr)# bandwidth 100
```

Syntax: [no] bandwidth <*rate*>

The <*rate*> variable specifies the bandwidth in Kilobits/sec for the bypass route.

Acceptable value can be between 0 and 2 Gigabits/sec.

A value of 0 means that the detour route uses a best-effort value for bandwidth.

The default value is 0.

Configuring a hop limit for a MPLS Fast Reroute

By default, a detour route can consist of no more than 255 hops. You can optionally change this maximum to a lower number.

For example, to limit any detour route in the LSP named "frr_tunnel" to no more than 20 hops.

```
NetIron(config-mpls-lsp-frr_tunnel)# frr
NetIron(config-mpls-lsp-frr_tunnel-frr)# hop-limit 20
```

Syntax: [no] hop-limit <*number*>

The number of hops can be from 0 – 255.

Configuring priority for a MPLS Fast Reroute

You can specify setup and hold priorities for the detour routes within a specified LSP. These priorities are available to any LSP and function exactly the same on standard LSPs as they do on detour LSPs. The priority determines the relative importance of the detour routes during setup or preemption. The priority has two components the setup priority and the hold priority.

If a detour LSP is assigned a higher setup priority, it can preempt any LSP (detour or otherwise) that is already established and has a lower holding priority, causing resources assigned to the lower priority LSP to be diverted to the higher priority LSP. The hold priority specifies how likely an established LSP is to give up its resources to another LSP. To be preempted, an LSP must have a lower hold priority than the preempting LSP 's setup priority. In addition, an established LSP can be preempted by a higher priority LSP only if it would allow the higher priority LSP to be established successfully.

To configure the detour routes of LSP frr_tunnel with a setup priority of 6 and hold priority of 1.

```
NetIron(config-mpls-lsp-frr_tunnel)# frr
NetIron(config-mpls-lsp-frr_tunnel-frr)# priority 6 1
```

Syntax: [no] priority <*setup-priority*> <*hold-priority*>

Possible values are 0 (highest priority) through 7 (lowest priority). A setup priority must be lower than or equal to the configured hold priority on an LSP. By default, the setup priority is 7 and the hold priority is 0.

Including or excluding administrative groups from LSP calculations

Administrative groups, also known as resource classes or link colors, allow you to assign MPLS-enabled interfaces to various classes. When a device calculates the path for a detour LSP, it takes into account the administrative group to which a interface belongs; you can specify which administrative groups the device can include or exclude when making its calculation.

For example, to include interfaces in either administrative group “gold” or “silver” in the path calculations for detour routes of the LSP frr_tunnel.

```
NetIron(config-mpls-lsp-frr_tunnel)# frr
NetIron(config-mpls-lsp-frr_tunnel-frr)# include-any gold silver
```

Syntax: [no] include-any <*groups*>

The value specified for <*groups*> can be one or more valid administrative group names or numbers. In this example, the device includes any of the interfaces that are members of groups “gold” or “silver” when calculating detour routes for this LSP. Only those interfaces in the “gold” or “silver” groups are considered for the detour routes. Interfaces that are not part of these groups, as well as interfaces that are not part of any group, are eliminated from consideration.

To exclude interfaces in either administrative group “gold” or “silver” when detour routes for LSP frr_tunnel are calculated.

```
NetIron(config-mpls-lsp-frr_tunnel)# frr
NetIron(config-mpls-lsp-frr_tunnel-frr)# exclude-any gold silver
```

Syntax: [no] exclude-any <*groups*>

In this example, the device excludes any of the interfaces that are members of groups “gold” or “silver” when calculating detour routes for this LSP. Only interfaces that are not part of either group can be considered for the detour routes.

To specify that an interface must be a member of both the “gold” or “silver” administrative groups in order to be included in the detour routes for LSP frr_tunnel.

```
NetIron(config-mpls-lsp-frr_tunnel)# frr
NetIron(config-mpls-lsp-frr_tunnel-frr)# include-all gold silver
```

Syntax: [no] include-all *<groups>*

In this example, an interface must be a member of all the groups specified in the **include-all** command to be considered in a detour route for the LSP. Any interface that is not a member of all the groups is eliminated from consideration.

Protecting MPLS LSPs through a bypass LSP

Implementing a bypass LSP to back up one or more MPLS LSPs requires the following tasks:

- The LSPs that you intend to have the protection of a bypass LSP must be enabled for Fast Reroute and then must be specified as needing facility backup. (You do not need to create the LSP before the bypass LSP is created because the bypass LSP identifies the LSPs to protect by interface IDs, not by LSP names.)
- An LSP is configured to be a bypass LSP with enough bandwidth for all the LSPs that it protects.
- The interfaces that get the protection of a bypass LSP are identified to that particular LSP. Protected LSPs can be identified by individual interfaces, ranges of interfaces, interface groups, or a LAG.

NOTE
The name of the bypass LSP must be unique among all bypass LSPs and all protected LSPs.

The sections that follow describe the items unique to the bypass LSP feature. The common LSP parameters are described elsewhere throughout this chapter.

Specifying an LSP to request facility backup

LSP mlxe3-199 is configured for Fast Reroute and then configured to request facility backup.

```
NetIron(config-mpls)# lsp mlxe3-199
NetIron(config-mpls-mlxe3-199)# frr
NetIron(config-mpls-mlxe3-199-frr)# facility-backup
```

Syntax: [no] facility-backup *<name>*

A subsequent iteration of the **show** command in the bypass LSP context shows that this LSP is a candidate for protection by a bypass LSP. The display for protected LSP mlxe3-199 shows that, under frr, the facility-backup line shows this protection is requested.

```
NetIron(config-mpls-bypasslsp-123)#show mpls config lsp mlxe3-199
lsp mlxe3-199
to 33.33.33.33
primary mlxe3-100
priority 4 3
secondary mlxe3-101
   standby
frr
   facility-backup
revert-timer 10
enable
```

Syntax: show mpls configuration lsp <*name*>

Specifying a bypass LSP

You can create a bypass LSP by using the bypass-lsp command. Thereafter, in the bypass LSP context, you must specify at least one interface as an exclude (protected) interface. This interface can be on a LAG. In this example, xm4 is specified to be a bypass LSP; the protected LSP interfaces are specified; and then the options for a bypass LSP are displayed.

```
NetIron(config)#router mpls
NetIron(config-mpls)#bypass-lsp xm4-by
NetIron(config-mpls-bypasslsp-xm4-by)#
NetIron(config-mpls-bypasslsp-xm4-by>)# exclude-interface e 1/2, e 1/2, e 2/5-e 2/
NetIron(config-mpls-bypasslsp-xm4-by>)#?
clear                          Clear table/statistics/keys
cos                            Class of service
disable                        Tear down the LSP
enable                         Establish the LSP
end                            End Configuration level and go to Privileged level
exclude-any                    Exclude any of the administrative groups
exclude-interface              choose the interface to avoid as well as protect
exit                           Exit current level
from                           Set ingress router of the LSP
hop-limit                      Limit of hops the LSP can traverse
include-all                    Include all of the administrative groups
include-any                    Include any of the administrative groups
metric                         Set the LSP metric
no                             Undo/disable commands
primary-path                   Set primary explicit path
priority                       Setup/hold priorities
quit                           Exit to User level
record                         Enable or disable recording path routes
show                           Display system information
tie-breaking                   Choose the tie breaking mode for cspf
to                             Set egress router of the LSP
traffic-eng                    Set traffic engineering parameters
write                          Write running configuration to flash or terminal
```

Syntax: [no] bypass-lsp <*name*>

The <*name*> must be unique among all regular LSPs and bypass LSPs.

Syntax: [no] exclude-interface <*linkid*>, <*linkid*>, <*linkid-begin-linkid-end*>

Syntax: [no] exclude-any <*group*>

Displaying MPLS and RSVP information

You can display the following information about the MPLS configuration on the device:

- Information about MPLS-enabled interfaces on the device
- Statistics about the MPLS-enabled interfaces, including bypass LSPs
- MPLS summary information
- Contents of the Traffic Engineering Database (TED)
- Status information about signalled LSPs configured on the device
- Information about paths configured on the device
- The label applied at each hop in an LSP
- Contents of the MPLS routing table
- RSVP information, including the status of RSVP-enabled interfaces, session information, and statistics
- Information about OSPF-TE LSAs
- MPLS Fast Reroute Information
- MPLS Bypass LSP

Displaying information about MPLS-enabled interfaces

To display information about the interfaces on the device that have been enabled for MPLS.

```
NetIron#show mpls interface ethernet 4/15
e4/15
  Admin: Up  Oper: Up
  Maximum BW: 1000000 kbps, maximum reservable BW: 1000000 kbps
  Admin group: 0x00000000
  Reservable BW [priority] kbps:
    [0] 780000    [1] 780000    [2] 780000    [3] 760000
    [4] 760000    [5] 760000    [6] 760000    [7] 760000
  Last sent reservable BW [priority] kbps:
    [0] 780000    [1] 780000    [2] 780000    [3] 760000
    [4] 760000    [5] 760000    [6] 760000    [7] 760000
  Configured Protecting bypass lsps:
mlxe4-by(UP)
```

Syntax: show mpls interface [ethernet <*slot/port*> **| pos** <*slot/port*> **| ve** <*vid*> **]**

For each MPLS-enabled interface on the device, the following information is displayed.

TABLE 198 Output from the show mpls interface command

This Field...	Displays...
Interface	The interface type refers to any one of the following: • **ethernet** <*slot/port*> to limit the display to a single ethernet port. • **pos** <*slot/port*> to limt the display to a single pos port. • **ve** <*vid*> to limit the display to a VE interface ID specified by the <*vid*> variable.
Maximum BW	The maximum outbound bandwidth that can be used on the interface. This TLV reflects the actual physical bandwidth of the interface.
maximum reservable BW	The maximum bandwidth that can be reserved on the interface. By default, the Maximum Reservable Bandwidth is the same as the Maximum Bandwidth for the interface. You can optionally change the reservable bandwidth to an amount greater than or equal to the maximum available bandwidth of the interface with the **traffic-eng reservable-bw** command.
Admin group	The administrative groups to which this interface belongs, set with the **admin-group** command.
Reservable BW [priority] kbps	The amount of bandwidth not yet reserved on the interface. Eight octets are displayed, indicating the amount of unreserved bandwidth (in kbits per second) that can be reserved with a hold priority of 0 through 7. The value in each of the octets is less than or equal to the maximum reservable bandwidth.
Last sent reservable BW [priority] kbps	The values in the Unreserved Bandwidth TLV sent in the most recent OSPF-TE LSA. If the device is not sending out OSPF-TE LSAs for the interface, the unreserved bandwidth value for each of the priorities is 0.
Configured Protecting bypass LSPs	The name and operational state of any bypass LSPs that are protecting this interface.

Displaying MPLS statistics

The following sections describe the commands used to gather MPLS statistics.

Displaying MPLS label statistics

To display all of the MPLS traffic statistics by their MPLS label, enter the following command.

```
NetIron# show mpls statistics label
In-label   In-Port(s)   In-Packet Count
    1024         e3/1            315431
                 e3/2            349193
                 e3/3                 0
                 e3/4                 0
    1025         e3/1            419750
                 e3/2                 0
                 e3/3                 0
                 e3/4                 0
    1024 e5/1 - e5/10            364690
         e5/11 - e5/20                0
         e5/21 - e5/30                0
    1025 e5/1 - e5/10                 0
         e5/11 - e5/20                0
         e5/21 - e5/30                0
```

When applicable, the **show mpls statistics label** command also displays statistics for LDP over RSVP traffic as displayed in the following example.

```
NetIron#show mpls statistics label
In-label In-Port(s)    In-Packet Count
1024    e2/1 - e2/2    45454
        e2/3 - e2/4     0
1025    e2/1 - e2/2     0
        e2/3 - e2/4   454528
1024    e4/1 - e4/20    0
1025    e4/1 - e4/20    0
```

To display all of the MPLS traffic statistics by their MPLS label for the PowerConnect B-MLXe, enter the following command.

```
PowerConnect# show mpls statistics label
In-label In-Port(s) In-Bytes Count
1024 e1/1-e1/24      315431
       e1/25-e1/48     0
```

To display all MPLS traffic statistics by their MPLS label that are gathered by the corresponding network processor that contains a specific port, enter a command such as the following.

```
NetIron# show mpls statistics label 3/1
In-label      In-Port(s)    In-Packet Count
    1024    e3/1 - e3/20                30
    1026    e3/1 - e3/20                21
    1030    e3/1 - e3/20               100
    1032    e3/1 - e3/20                 0
    1033    e3/1 - e3/20                 0
    1034    e3/1 - e3/20                12
    1036    e3/1 - e3/20                 0
```

To display all MPLS traffic statistics by their MPLS label for a specific port for the PowerConnect B-MLXe, enter a command such as the following.

```
PowerConnect# show mpls statistics label 1/1
In-label In-Port(s) In-Bytes Count
1024 e1/1-e1/24      315431
```

Table 199 lists the output displayed for the **show mpls statistics label** command.

TABLE 199 MPLS statistics label parameters

This field...	Displays...
In-label	The MPLS label ID.
In-Port (s)	The port where the traffic arrives.
In-Packet Count	The number of packets meeting the In-label and In-port criteria.
In-Bytes Count	The number of bytes meeting the In-label and In-port criteria.

Syntax: show mpls statistics label <*interface*>

The <*interface*> variable allows you to limit label statistics displayed to a specified interface.

Displaying MPLS tunnel statistics

To display all of the MPLS traffic statistics by their MPLS tunnel, enter the following command.

```
NetIron# show mpls statistics tunnel
Tunnel       In-Port(s)       L3VPN/IPoMPLS Out-Pkt
0            e1/1 - e1/20     0
             e2/1 - e2/2      0
             e3/1 - e3/2      0
             e3/3 - e3/4      0
             e4/1 - e4/20     0
1            e1/1 - e1/20     0
             e2/1 - e2/2      0
             e3/1 - e3/2      0
             e3/3 - e3/4      0
             e4/1 - e4/20     0
```

To display all of the MPLS traffic statistics for a specific tunnel, enter a command such as the following.

```
NetIron# show mpls statistics tunnel 1
Tunnel       In-Port(s)       L3VPN/IPoMPLS Out-Pkt
1            e1/1 - e1/20     0
             e2/1 - e2/2      0
             e3/1 - e3/2      0
             e3/3 - e3/4      0
             e4/1 - e4/20     0
```

Table 200 lists the output displayed for the **show mpls statistics tunnel** command.

TABLE 200 MPLS statistics tunnel parameters

This field...	Displays...
Tunnel	The index number of the MPLS tunnel.
In-Port(s)	The port where the traffic is received.
L3VPN/IPoMPLS Out-Pkt	The number of Layer 3 VPN and IPoMPLS packets that have been sent outbound meeting the In-label and Tunnel criteria.

Syntax: show mpls statistics tunnel <*tunnel-index*>

The <*tunnel-index*> variable allows you to limit MPLS statistics displayed to a specified tunnel.

NOTE
When the traffic is being forwarding using PBR over MPLS tunnel, the MPLS tunnel statistics will not be incremented and only the PBR accounting will indicate these packets in the statistics.

Displaying MPLS VRF statistics

To display out-packet statistics for VRFs, enter the following command.

```
NetIron# show mpls statistics vrf
VRF Name        In-Port(s)     Endpt Out-Pkt     Tnl Out-Pkt
red             e3/1           0                 0
                e3/2           0                 0
                e3/3           0                 0
                e3/4           0                 0
                e5/1 - e5/10   0                 0
                e5/11 - e5/20  0                 0
                e5/21 - e5/30  0                 0
                e5/31 - e5/40  0                 0
green           e3/1           3707480           0
                e3/2           2692915           0
                e3/3           0                 0
                e3/4           0                 0
                e5/1 - e5/10   0                 0
                e5/11 - e5/20  0                 5834179
                e5/21 - e5/30  0                 0
                e5/31 - e5/40  0                 0
pink            e3/1           0                 0
                e3/2           0                 0
                e3/3           0                 0
                e3/4           0                 0
                e5/1 - e5/10   0                 0
                e5/11 - e5/20  0                 0
                e5/21 - e5/30  0                 0
                e5/31 - e5/40  0                 0
```

To display out-packet statistics for a specific VRF, enter the following command.

```
NetIron# show mpls statistics vrf black
VRF Name        In-Port(s)     Endpt Out-Pkt     Tnl Out-Pkt
black           e3/1           0                 0
                e3/2           29607351          0
                e3/3           27522998          25828420
                e3/4           0                 0
                e5/1 - e5/10   0                 0
                e5/11 - e5/20  0                 0

                e5/21 - e5/30  0                 0
                e5/31 - e5/40  0                 0

            e5/31 - e5/40  0
```

Table 201 lists the output for the **show mpls statistics vrf** command.

TABLE 201 MPLS statistics VRF parameters

This field...	Displays...
VRF Name	The name of the VRF from which packets originated or are destined.
In-Port(s)	The port that is either the VRF or MPLS interface.
Endpt Out-Pkt	The number of packets forwarded to the specified VRF interface.
Tnl Out-Pkt	The number of VRF data packets sent to the remote peer over an MPLS tunnel.

Syntax: show mpls statistics vrf <*vrf-name*>

The <*vrf-name*> variable allows you to limit the display of VRF statistics to a specific VRF.

Displaying LSP accounting statistics

If the **ingress-tunnel-accounting** command has been configured, you can display accounting statistics for RSVP-signaled LSPs using the commands described in this section.

If an LSP has the protection of a bypass LSP (as described in "MPLS Fast Reroute using facility backup over a bypass LSP" on page 1311), statistics for the protected LSP will always be shown under the protected LSP even if the protected LSP switches over to the backup LSP riding on a bypass LSP. Bypass LSP is not listed in the output of the **show mpls statistics lsp** command. Also, if you request statistics and name a bypass LSP, the system generates an error message.

For example, assume that LSP1 and LSP2 are protected LSPs and that BYPASS is the bypass LSP that protects both LSP1 and LSP2.

```
NetIron# show mpls statistic lsp BYPASS
Error - cannot find lsp BYPASS
```

If the bypass LSP is implicit NULL, then statistics are collected for a protected LSP only if the protected LSP is not implicit NULL.

Displaying LSP accounting statistics

To display RSVP-signaled LSP accounting statistics, enter the following command.

```
NetIron# show mpls statistics lsp
LSP tope4
  Tunnel index      0  0 pkt  0 Byte  0 Avg. pps  0 Avg. Bps
LSP 400
  Tunnel index      2  0 pkt  0 Byte  0 Avg. pps  0 Avg. Bps
LSP 4000
  Tunnel index      3  0 pkt  0 Byte  0 Avg. pps  0 Avg. Bps
LSP tope41
  Tunnel index      4  99205408 pkt  11314220016 Byte  84459 pps  9628340 Bps
```

When the **ingress-tunnel-accounting** command is enabled on the RSVP tunnel, the **show mpls statistics lsp** command displays statistics for LDP over RSVP traffic.

```
NetIron#show mpls statistics lsp
LSP test1
Tunnel interface tnl2 4241 pkt 1187480 Byte  10 pps  2800 Bps
LSP test2
Tunnel interface tnl3  0 pkt  0 Byte 0    Avg. pps   0 Avg. Bps
```

Syntax: show mpls statistics lsp [<*name*> **]**

The <*name*> variable specifies the LSP for which LSP accounting statistics are displayed. If you do not specify an LSP name, statistics are displayed for all RSVP-signalled LSPs.

Displaying LDP accounting statistics

To display LDP-signalled LSP accounting statistics, enter the following command.

```
NetIron# show mpls statistics ldp tunnel
LDP tunnel index  0  4241 pkt  1187480 Byte 10 Avg. pps  2800 Avg. Bps
```

Syntax: show mpls statistics ldp tunnel [<*tunnel-index*> **]**

The <*tunnel-index*> variable specifies the index number of the MPLS tunnel for which you want to display LDP-signalled, LSP accounting statistics. If you do not specify an index number, LSP accounting statistics are displayed for all LDP-signalled LSPs.

Table 202 shows the output of the **show mpls statistics lsp** and **show mpls statistics ldp tunnel** commands.

TABLE 202 MPLS LSP statistics parameters

This field...	Displays...
LSP	The name of the LSP that statistics are being displayed for. (Displayed for RSVP-signaled LSPs only.)
Tunnel	The index number of the MPLS tunnel.
pkt	The total number of packets forwarded through the specified LSP.
Byte	The total number of bytes forwarded through the specified LSP.
Avg. pps	The number of packets-per-second forwarded through the specified LSP.
Avg. Bps	The number of bytes-per-second forwarded through the specified LSP.

NOTE
The LSP accounting feature will display the primary tunnel interface even when the traffic is sent through the secondary path.

Clearing LSP and LDP accounting statistics

Byte and packet counters can be cleared for RSVP-signaled LSPs using the following command.

Syntax: clear mpls statistics lsp [<*name*>]

The <*name*> variable specifies LSP that you want to clear byte and packet counters for. If you do not specify an LSP name, byte and packet counters will be cleared for all RSVP-signalled LSPs.

Byte and packet counters can be cleared for LDP-signaled LSPs using the following commands.

Syntax: clear mpls statistics ldp tunnel [<*tunnel-index*>]

The <*tunnel-index*> variable specifies the index number of the MPLS tunnel for which you want to clear byte and packet counters. If you do not specify an index number, byte and packet counters are cleared for all LDP-signalled LSPs.

Displaying MPLS summary information

You can display a summary of MPLS information, including the number of configured paths and signalled LSPs for which this device is the ingress LER.

Example

```
NetIron# show mpls summary

Path:
        Paths configured    =     0

Signaled LSPs:
        LSPs configured     =     1
        LSPs enabled        =     1
        LSPs operational    =     0
```

Syntax: show mpls summary

Displaying the Traffic Engineering database

An LSR's Traffic Engineering Database (TED) contains topology information about nodes in an MPLS domain and the links that connect them. This topology information is obtained from either the OSPF traffic engineering (OSPF-TE) LSAs or IS-IS LSPs with traffic engineering extensions. OSPF-TE LSAs and IS-IS LSPs with TE extensions have special extensions that contain information about an MPLS-enabled interface's traffic engineering metric, bandwidth reservations, and administrative group memberships.

An LSR, when configured to do so, floods OSPF-TE LSAs or IS-IS LSPs with TE extensions for its MPLS-enabled interfaces to its neighboring routers in the OSPF or IS-IS area. Other LSRs store the information from the OSPF-TE LSAs or IS-IS LSPs with TE extensions in their own Traffic Engineering Databases, allowing each LSR in the area to maintain an identical TED describing the MPLS topology. The topology information in the TED is used by the CSPF process when it calculates traffic-engineered paths for signalled LSPs.

You can display the contents of an LSR's TED. The following example is for a router where OSPF-TE LSAs are enabled for MPLS interfaces.

```
NetIron# show mpls ted database
AreaId: 0
  NodeID: 2.2.2.2, Type: Router
    Type: M/A, To: 10.1.1.3, Local: 10.1.1.2, Remote: 0.0.0.0
  NodeID: 3.3.3.3, Type: Router
    Type: P2P, To: 6.6.6.6, Local: 40.1.1.1, Remote: 40.1.1.2
    Type: M/A, To: 10.1.1.3, Local: 10.1.1.3, Remote: 0.0.0.0
    Type: M/A, To: 20.1.1.2, Local: 20.1.1.1, Remote: 0.0.0.0
  NodeID: 10.1.1.3, Type: Network
    Type: M/A, To: 1.1.1.1, Local: 0.0.0.0, Remote: 0.0.0.0
    Type: M/A, To: 2.2.2.2, Local: 0.0.0.0, Remote: 0.0.0.0
    Type: M/A, To: 3.3.3.3, Local: 0.0.0.0, Remote: 0.0.0.0
  NodeID: 30.1.1.2, Type: Network
    Type: M/A, To: 1.1.1.1, Local: 0.0.0.0, Remote: 0.0.0.0
    Type: M/A, To: 6.6.6.6, Local: 0.0.0.0, Remote: 0.0.0.0
```

The following example is for a router where IS-IS TE LSPs are enabled for MPLS interfaces.

```
NetIron# show mpls ted database
This Router is MLXe3
Global Link Gen 106
ISIS(2)
  NodeID: MLXe4.00(12.12.12.12) , Type: Router
    Type: M/A, To: MLXe3.02 Local: 122.0.0.1, Remote: 0.0.0.0, Gen 54
    Type: M/A, To: MLXe3.03 Local: 129.0.0.2, Remote: 0.0.0.0, Gen 51
    Type: M/A, To: PE2-JNPR.04 Local: 147.0.0.1, Remote: 0.0.0.0, Gen 57
    Type: M/A, To: GSR.02 Local: 125.0.0.1, Remote: 0.0.0.0, Gen 78
  NodeID: MLXe3.00(15.15.15.15) , Type: Router
    Type: M/A, To: MLXe3.02 Local: 122.0.0.2, Remote: 0.0.0.0, Gen 49
    Type: M/A, To: MLXe3.03 Local: 129.0.0.1, Remote: 0.0.0.0, Gen 50
  NodeID: PE2-JNPR.00(5.5.5.5) , Type: Router
    Type: M/A, To: PE2-JNPR.04 Local: 147.0.0.2, Remote: 0.0.0.0, Gen 62
    Type: M/A, To: PE2-JNPR.02 Local: 126.0.0.1, Remote: 0.0.0.0, Gen 60
    Type: M/A, To: PE4.03 Local: 148.0.0.2, Remote: 0.0.0.0, Gen 104
  NodeID: GSR.00(9.9.9.9) , Type: Router
    Type: P2P, To: PE4.00(14.14.14.14) Local: 128.0.0.2, Remote: 128.0.0.1, Gen 1
    Type: M/A, To: GSR.02 Local: 125.0.0.2, Remote: 0.0.0.0, Gen 65
    Type: M/A, To: PE2-JNPR.02 Local: 126.0.0.2, Remote: 0.0.0.0, Gen 77
```

Syntax: show mpls ted database

The following table describes the output of the **show mpls ted database** command.

TABLE 203 Output from the show mpls ted data command

This field...	Displays...
Global Link Gen	The number of times the TED database has changed.
AreaId	The ID of this OSPF area.
NodeID	The ID of the node. For Router nodes, can be any interface address or a loopback interface address on the LSR. For Network nodes, this is the router ID of the network's designated router.
[node] Type	The type of node. The node type can be either Router or Network RouterIndicates the node is an actual LSR. NetworkIndicates the node represents a multi-access network.
[link] Type	The type of link. The link type can be either P2P or M/A P2P Indicates this is a point-to-point link. M/A Indicates the link is a broadcast, multi-access network.
To	The ID of the node at the end of this link.
Local	The address of the interface used to reach the remote node.
Remote	The address of the interface on the remote node that is connected to the local node. For M/A link types, this is always 0.0.0.0.
Gen	The number of times this link information has changed.

To display more detailed information about each node in the TED.

```
NetIron# show mpls ted database detail
AreaId: 0
  NodeID: 2.2.2.2, Type: Router
    Type: M/A, To: 10.1.1.3, Local: 10.1.1.2, Remote: 0.0.0.0
      Color: 0x00000007
      Metric: 1
      Max BW: 155000 kbps
      Reservable BW: 155000 kbps
      Available BW [priority] kbps:
        [0] 155000       [1] 155000       [2] 155000       [3] 155000
        [4] 155000       [5] 155000       [6] 155000       [7] 155000
  NodeID: 1.1.1.1, Type: Router
    Type: M/A, To: 10.1.1.3, Local: 10.1.1.1, Remote: 0.0.0.0
      Color: 0x00000007
      Metric: 1
      Max BW: 155000 kbps
      Reservable BW: 155000 kbps
      Available BW [priority] kbps:
        [0] 155000       [1] 155000       [2] 155000       [3] 155000
        [4] 155000       [5] 155000       [6] 155000       [7] 155000
    Type: M/A, To: 30.1.1.2, Local: 30.1.1.1, Remote: 0.0.0.0
      Color: 0x00000007
      Metric: 1
      Max BW: 155000 kbps
      Reservable BW: 155000 kbps
      Available BW [priority] kbps:
        [0] 155000       [1] 155000       [2] 155000       [3] 155000
        [4] 155000       [5] 155000       [6] 155000       [7] 155000
```

Syntax: show mpls ted database detail

In addition to the information described in Table 203, the **show mpls ted database detail** command gives the following

TABLE 204 Output from the show mpls ted database detail command

This field...	Displays...
Color	The administrative groups to which this interface belongs.
Metric	The traffic engineering metric for the interface (by default, this is equal to the OSPF link cost).
Max BW	The maximum outbound bandwidth that can be used on the interface. This is the actual physical bandwidth of the interface (155M for OC-3, 622M for OC-12, or 2488M for OC-48).
Reservable BW	The maximum bandwidth that can be reserved on the interface. By default, the Maximum Reservable Bandwidth is the same as the Maximum Bandwidth for the interface.
Available BW [priority] kbps	The amount of bandwidth not yet reserved on the interface. Eight octets are displayed, indicating the amount of unreserved bandwidth (in kbits per second) that can be reserved with a hold priority of 0 through 7. The value in each of the octets is less than or equal to the maximum reservable bandwidth.

Displaying a traffic engineering path to a destination

You can display a traffic engineering path to a IPv4 destination address using a specified set of resource parameters. This enhancement allows you to gain insight into a traffic engineering path in a network, before setting it up using RSVP. This will help you to avoid RSVP path setup failure due to unavailable requested resources along the path to the destination host.

To display the traffic engineering path to a IPv4 destination address, enter the following command.

```
NetIron# show mpls ted path 4.4.4.4
```

Syntax: **show mpls ted path** <*destIPAddress*> [**bandwidth** <*bw_in_kbps*>] [**hop-limit** <*max_hops*>] [**priority** <*setup_priority*>] [**exclude-any** <*STRING_List_of_Admin_Resource_groups_name_or_and_num*>] [**include-any** <*STRING_List_of_Admin_Resource_groups_name_or_and_num*>] [**include-all** <*STRING_List_of_Admin_Resource_groups_name_or_and_num*>] [**tie-breaking {random | least-fill | most-fill}**]

NOTE
When configuring the CLI command, any combination of resource constraint parameters can be used.

The following table describes the parameters of the **show mpls ted path** command

TABLE 205 Parameters from the show mpls ted path command

CLI parameter	Description
<*destIPAddress*>	The IPv4 address of the destination host.
bandwidth <*bw_in_kbps*>	The minimum bandwidth of the path to its destination. The bandwidth value is entered in decimal in kilo bits per second unit. The valid range is between 0- 2147483647. If the value entered is larger than 2147483647, then the value will be truncated to the max limit of 2147483647 and accepted as the bandwidth input.
hop-limit <*max_hops*>	The maximum hops for the path to reach to its destination. The valid range is between 0 - 255. If an invalid range is entered, then an error message will display. If a path to the destination is available, but the hop count for the path is greater than <*max_hops*> value, then MPLS will indicate that path is not available.
priority <*setup_priority*>	The setup priority of the path. The valid range is between 0 - 7. The default is 7, the lowest setup priority value. If an invalid range is entered, then an error message will display. The priority parameter should be entered along with the bandwidth parameter because while setting up an LSP, the setup priority value decides the ability to reserve a bandwidth amount.
exclude-any <*STRING*>	The <*STRING*> variable specifies the admin group name or number. The string must be enclosed in double quotes.The <*STRING*> variable is a list of any combination of admin group name and number. The valid range for the admin group number is between 0 and 31. The admin group name must start with an alphabet character. If an invalid range is entered for admin group number and admin group name, then the CLI will prompt a warning message. The CLI will be accepted, but the out of range value will be ignored.
include-any <*STRING*>	The <*STRING*> variable specifies the admin group name or number. The string must be enclosed in double quotes.The <*STRING*> variable is a list of any combination of admin group name and number. The valid range for the admin group number is between 0 and 31. The admin group name must start with an alphabet character. If an invalid range is entered for admin group number and admin group name, then the CLI will prompt a warning message. The CLI will be accepted, but the out of range value will be ignored.

TABLE 205 Parameters from the show mpls ted path command (Continued)

CLI parameter	Description
include-all <*STRING*>	The <*STRING*> variable specifies the admin group name or number. The string must be enclosed in double quotes.The <*STRING*> variable is a list of any combination of admin group name and number. The valid range for the admin group number is between 0 and 31. The admin group name must start with an alphabet character. If an invalid range is entered for admin group number and admin group name, then the CLI will prompt a warning message. The CLI will be accepted, but the out of range value will be ignored.
tie-breaking {random | least-fill | most-fill}	The tie-breaking method is used when multiple equal cost paths to the destination exist. The tie-breaking rule will select only one path to be displayed from among multiple equal cost paths. The default is random.

The following example displays an output from the **show mpls ted path** command.

```
NetIron# show mpls ted path 12.12.12.12 hop-limit 2
Path to 12.12.12.12 found! Time taken to compute: 0 msec
Hop-count: 2 Cost: 2000 ISIS Level-1
        Hop 1: 40.1.0.1, Rtr 13.13.13.13
        Hop 2: 50.1.0.2, Rtr 12.12.12.12
```

The following is an example for a router where the exclude-any parameter is used.

```
NetIron#show mpls ted path 11.11.11.11 exclude-any 0
Path to 11.11.11.11 found! Time taken to compute: 0 msec
Hop-count: 1 Cost: 10 ISIS Level-2
        Hop 1: 129.0.0.13, Rtr 11.11.11.11
```

The following is an example of an output with an error message using the hop-limit parameter, when an out of range parameter value is entered.

```
NetIron(config-mpls)# show mpls ted path 2.2.2.2 hop-limit 300
Error - Hop count value is out of range [0 - 255]
```

If an out of range parameter value is entered, the following error message is displayed for the priority parameter.

Priority

Error - Setup priority value is out of range [0 - 7]

The following table describes the output of the **show mpls ted path** command

TABLE 206 Output from the **show mpls ted path** command

This field...	Displays...
Path to 4.4.4.4 found	The IPv4 address of the destination host is found.
Time taken to compute	The total time taken by CSPF (in milliseconds) to compute this path.
Hop-count	The hop count of this path.
Cost	The total cost of this path.
ISIS	The ISIS or OSPF or CSPF area ID through which this path will traverse.

TABLE 206 Output from the **show mpls ted path** command

This field...	Displays...
Hop	The ingress interface IPv4 address at each hop.
Rtr	The traffic engineering router ID (IPv4 address) at each hop.

Displaying signalled LSP status information

You can display status information about signalled LSPs for which the device is the ingress LER as shown in the example below.

```
NetIron# show mpls lsp
*: The LSP is taking a Secondary Path
                                Admin Oper  Tunnel   Up/Dn Retry Active
Name            To              State State Intf     Times No.   Path
t1              3.3.3.3         UP    UP*   tnl1     1     5     v2
```

Syntax: show mpls lsp [brief]

NOTE
The **show mpls lsp brief** command displays the same information as the **show mpls lsp** command.

Table 207 describes the output of the **show mpls lsp** command.

TABLE 207 Output from the **show mpls lsp** command

This field...	Displays...
Name	The name of the LSP. LSPs are displayed in alphabetical order.
To	The egress LER for the LSP.
Admin State	The administrative state of the LSP. Once you activate the LSP with the **enable** command, the administrative state changes from DOWN to UP.
Oper State	The operational state of the LSP. This field indicates whether the LSP has been established through signalling and is capable of having packets forwarded through it. There may be a short period of time after you enable the LSP that the administrative state of the LSP is UP, but the operational state is DOWN. Once the LSP has been established through signalling, both the administrative state and the operational state will be UP.
Tunnel Intf	The MPLS tunnel interface port ID.
Up or Dn Times	The number of times the operational state of the LSP's primary path has transitioned from DOWN to UP.
Retry No.	The number of attempts the ingress LER has made to connect to the egress LER.
Active Path	The path currently in use for this LSP. Dashes (--) indicate that there is no named path for the LSP or that the LSP has not yet been established over the named path.

The **show mpls lsp detail** command displays detailed information about a specific LSP. To display detailed information about the status of the LSPs for which the device is the ingress LER, enter the **show mpls lsp detail** command as shown in the following example.

```
NetIron# show mpls lsp detail
LSP t1, to 3.3.3.3
  Path selected: pathsecondaryb, mode: manual revert-timer:
  Path selected is up for 3 seconds for the latest instance, traffic will be switchec
  it in 7 seconds.
  From: 1.2.3.4, admin: UP, status: UP, tunnel interface: tnl1
  Times primary LSP goes up since enabled: 1
  Metric: 1, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 3
  Pri. path: dir, active: no
   Setup priority: 7, hold priority: 0, ipmtu 1400
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   Tie breaking: random, hop limit: 0
  Sec. path: v2, active: active
   Hot-standby: no, status: up
   Setup priority: 7, hold priority: 0
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl1, outbound interface: e1/1
   Tunnel index: 5, outbound label: 1966
   Path calculated using constraint-based routing: no
   Explicit path hop count: 1
    10.10.10.2 (S) -> 20.20.20.2 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    10.10.10.2 -> 20.20.20.2
   BFD session status: UP
    Config param: global, min-tx: 1000, min-rx: 1000, multiplier: 3
    Negotiated tx-interval: 1000, rx-interval: 3000, multiplier: 3
    Local discriminator: 1, remote discriminator: 1
```

Syntax: show mpls lsp [detail | <*lsp_name*>]

The <*lsp_name*> variable specifies the name of the LSP you want to display.

Table 208 describes the output from the **show mpls lsp detail** command.

TABLE 208 Output from the **show mpls lsp detail** command

This field...	Displays...
Name	The name of the LSP. LSPs are displayed in alphabetical order.
To	The egress LER for the LSP.
From	The LSP's source address, configured with the **from** command. If a source IP address has not been specified for the LSP with the **from** command, and the LSP has not been enabled, then "(n/a)" is displayed in the From field.
admin	The administrative state of the LSP. Once you activate the LSP with the **enable** command, the administrative state changes from DOWN to UP.
status	The operational state of the LSP. This field indicates whether the LSP has been established through signalling and is capable of having packets forwarded through it. If the status of the LSP is DOWN, the reason why the LSP is down is shown in parentheses. There may be a short period of time after you enable the LSP that the administrative state of the LSP is UP, but the status is DOWN. Once the LSP has been established through signalling, both the administrative state and the status will be UP.

TABLE 208 Output from the **show mpls lsp detail** command (Continued)

This field...	Displays...
Path selected	The path currently selected for this LSP.
mode	The path selection mode currently configured for the LSP. It can be one of the following: • auto • manual • unconditional
revert-timer	The time in seconds for which the revert-timer has been configured.
Path selected is ...	The current status of the selected path. During a transition period, this message describes the number of seconds that the path has been up in the latest instance and the number of seconds before traffic will be switched to it.
Times primary LSP goes up	The number of times the status of the LSP's primary path has transitioned from DOWN to UP.
Metric	The metric for the LSP, configured with the **metric** command.
no. of installed aliases	The number of aliases that have been installed for the egress LER.
Max retries	The maximum number of attempts the ingress LER will attempt to connect to the egress LER, set with the **retry-limit** command.
no. of retries	The number of attempts the ingress LER has made to connect to the egress LER.
Pri. path	The name of the primary path for this LSP and whether the path is currently active.
Sec. path	The name of the secondary path for this LSP and whether the path is currently active.
Hot-standby	Whether the secondary path is a hot-standby path.
status	The operational state of the secondary path.
Setup priority	The configured setup priority for the LSP.
hold priority	The configured hold priority for the LSP.
ipmtu	The maximum packet size in bytes for IP packets transiting the LSP without being fragmented.
Max rate	The maximum rate of packets that can go through the LSP (in kbps), set with the **traffic-eng max-rate** command.
mean rate	The average rate of packets that can go through the LSP (in kbps), set with the **traffic-eng mean-rate** command.
max burst	The maximum size (in bytes) of the largest burst the LSP can send at the maximum rate, set with the **traffic-eng max-burst** command.
Constraint-based routing enabled	Whether CSPF is in effect for the LSP.
Tie breaking	The tie-breaking method CSPF uses to select a path from a group of equal-cost paths to the egress LER, set with the **tie-breaking** command.
hop limit	The maximum number of hops a path calculated by CSPF can have, set with the **hop-limit** command.
outbound interface	The outbound interface taken by the active path of the LSP. If the egress interface is a VE-enabled interface, the VE interface ID specified by the <*vid*> variable is displayed here in the Outbound interface field (for example, ve 20).
Tunnel index	The tunnel index for the active path of the LSP.
outbound label	The outbound label used by the active path of the LSP.

TABLE 208 Output from the **show mpls lsp detail** command (Continued)

This field...	Displays...
Path calculated using constraint-based routing	Whether the explicit path used by the active path was calculated using the constraint-based routing.
Explicit path hop counts	The number of explicit hops configured for the LSP, the addresses of the hops, and whether the hops are strict (S) or loose (L).
Recorded routes	The addresses recorded by the RECORD_ROUTE object during RSVP signalling.
If the BFD admin state is enabled for the LSP, the following BFD information will be displayed.	
BFD session status	Describes the status of the BFD session on this LSP as either UP or DOWN. If a BFD session for the LSP is enabled and the BFD session is DOWN, one of the following reasons for failure will be displayed: • BFD disabled globally • LSP down • Max session exceeded • Max LP session exceeded • Peer session down • Wait for peer
Config Param	Describes how the BFD configuration values were derived: • global – Displayed if the configuration values were derived from the router's global LSP BFD configuration. • local – Displayed if the configuration values were derived from a BFD configuration specific to this LSP.
min-tx	The min-tx value in milliseconds that is configured for this LSP.
min-rx	The min-rx value in milliseconds that is configured for this LSP.
multiplier	The multiplier value that is configured for this LSP.
tx-interval	The tx-interval value in milliseconds that has been negotiated between this router and its peer for this LSP.
rx-interval	The rx-interval value in milliseconds that has been negotiated between this router and its peer for this LSP.
multiplier	The multiplier value in milliseconds that has been negotiated between this router and its peer for this LSP.
Local discriminator	Value of the "local discriminator" field in the BFD Control Message as used by the local router in the last message sent.
remote discriminator	Value of the "local discriminator" field in the BFD Control Message as received in the last message sent by the remote peer.

The **show mpls lsp wide** command allows the user to display the full LSP name in a single line. Previously, a long LSP name (greater than 12 characters) was text-wrapped in multiple lines. Now, the full LSP name can be displayed in a single line as shown in the following example.

NOTE
The **show mpls lsp wide** command is supported on PowerConnect B-MLXe devices.

```
NetIron(config)#show mpls lsp wide
Note: LSPs marked with * are taking a Secondary Path
                                                        Admin Oper  Tunnel   Up/Dn Retry Active
Name                                                    To              State State Intf     Times No.   Path
tunnel1                                                 3.3.3.3         UP    UP    tnl0     1     0     --
tunnel2                                                 3.3.3.3         UP    UP    tnl4     1     0     ppath1
tunnelfromsanfranciscotonewyork                         3.3.3.3         UP    UP    tnl3     1     0
pathfromsanfranciscotonewyork
```

Syntax: show mpls lsp wide

The **include** option can be used with the **show mpls lsp wide** command to filter and display a specific LSP name.

```
NetIron#show mpls lsp wide | include tunnelfromsanfranciscotonewyork
                                                      Admin Oper  Tunnel   Up/Dn Retry Active
Name                                                   To              State State Intf     Times No.   Path
tunnelfromsanfranciscotonewyork                        3.3.3.3         UP    UP    tnl3     1     0
pathfromsanfranciscotonewyork
```

Syntax: show mpls lsp [**wide** [**| include** <*lsp_name*>]]

The <*lsp_name*> variable specifies the name of the LSP you want to display.

Displaying path information

A path is a list of router hops that specifies a route across an MPLS domain. You can create a path and then configure LSPs that see the path. When the LSP is enabled, the ingress LER attempts to signal the other LSRs in the path, so that resources can be allocated to the LSP.

You can display information about the paths configured on the device as shown in the following example.

```
NetIron#show mpls path
Path Name           Address             Strict/loose        Usage Count
to110_120           110.110.110.2       Strict              1
                    120.120.120.3       Strict
to2_pri             10.10.10.2          Strict              0
to2_sec             110.110.110.2       Strict              0
to3                 110.110.110.2       Loose               1
                    120.120.120.3       Loose
to3_pri             10.10.10.2          Strict              1
                    20.20.20.3          Strict
to3_sec             110.110.110.2       Strict              0
                    120.120.120.3       Strict
to4                 110.110.110.2       Loose               1
                    120.120.120.3       Loose
                    130.130.130.4       Loose
to_23               110.110.110.2       Strict              1
                    20.20.20.3          Strict
```

Syntax: show mpls path [<*path-name*>]

You can display information for all paths configured on the device, or for a specified <*path-name*>. Table 209 describes the output shown in the **show mpls path** command.

TABLE 209 Output of the **show mpls path** command

This column...	Displays...
Path Name	The configured name of the path.
Address	The IP address of each node in the path. A node corresponds to an MPLS-enabled router in the network.
Strict or loose	Whether the node is strict or loose. A strict node means that the router must be directly connected to the preceding node. A loose node means that other routers can reside between the source and destination nodes.
Usage Count	The number of LSPs that are either currently using or configured to use the path. For example, if an LSP named "to_sqa" has primary and secondary paths and both paths are configured to use the same MPLS path "path_to_sqa," then the usage count for "path_to_sqa" would be two (if no other LSP in the system is configured to use "path_to_sqa").

The **show mpls path wide** command allows the user to display the full path name in a single line. Previously, a long path name (greater than 12 characters) was text-wrapped in multiple lines. Now, the full path name can be displayed in a single line as shown in the following example.

NOTE
The **show mpls path wide** command is supported on PowerConnect B-MLXe devices.

```
NetIron(config)#show mpls path wide
Path Name                          Address       Strict/loose   Usage
Count
Pathfromsanfranciscotonewyork       10.10.10.2   Strict          1
ppath1                              10.10.10.2   Strict          1
spath1                              20.20.20.2   Strict          1
```

Syntax: show mpls path wide

The **include** option can be used with the **show mpls path wide** command to filter and display a specific path.

```
NetIron#show mpls path wide | include pathfromsanfranciscotonewyork
Path Name                          Address       Strict/loose   Usage
Pathfromsanfranciscotonewyork      10.10.10.2      Strict         1
```

Syntax: show mpls path [**wide** [**| include** <*path-name*>]]

The <*path-name*> variable specifies the name of the path you want to display.

Displaying the MPLS routing table

The MPLS routing table is used to store routes to egress LERs.

To display the contents of the MPLS routing table, enter the **show mpls route** command, as follows. The port field now displays whether an interface/port is either Ethernet or POS. For example, Ethernet port 3 on slot 2 is displayed as e2/3.

NOTE
The output display from the **show mpls route** command is applicable to PowerConnect B-MLXe devices.

```
NetIron#show mpls route
Total number of MPLS tunnel routes: 1
R:RSVP L:LDP S:Static O:Others
      Destination          Gateway           Tnnl    Port  Label  Sig Cost Use
1     140.140.140.4/32     140.140.140.4     tnl0    e1/2  3      R   0    0
```

Syntax: show mpls route [*<ip_prefix_addr>*/*<mask_len>* **[longer] |** *<ip_prefix_addr>* *<ip_mask>* **[longer]]**

where:

- *<ip_prefix_addr>* means display the route to the specified IP address.
- *<mask_len>* is the number of bits in the mask.
- **longer** means that if an IP address has been specified, display only the routes that match that IP address.
- *<ip_mask>* is the bytes of a network mask or, for CIDR format, the number of bits in the network mask.

TABLE 210 Output from the show mpls route command

This field...	Displays...
Destination	The destination for the route. This can be either the address of the egress LER in an LSP, or a configured alias.
NetMask	The network mask for the route. If the destination address is the egress LER, the mask is 32 bits.
Gateway	The address of the egress LER in the LSP. If the destination address is not a network alias, the gateway is the same as the destination address.
Port	The MPLS tunnel interface associated with the LSP. The port field displays whether an interface/port is an Ethernet port, POS port, or a VE interface. The VE interface ID is specified by the *<vid>* variable. When applicable, the egress interface of the routing entry displays the VE interface. The port display format for interface/port is as follows: • [e|p] <slot>/<port> "e" represents an Ethernet port "p" represents a POS port
Cost	The metric for the LSP, set with the **metric** command in the LSP's configuration.
Label	The MPLS label received from the downstream router.
sig	The signal protocol type associated with the label. Possible values are: • L – LDP • R – RSVP
Usage	The number of LSPs that are either currently using or configured to use the path. For example, if an LSP named "to_sqa" has primary and secondary paths and both paths are configured to use the same MPLS path "path_to_sqa," then the usage count for "path_to_sqa" would be two (if no other LSP in the system is configured to use "path_to_sqa")."

The **show mpls forwarding** command is introduced to display MPLS forwarding information. The out-intf field in the output of the **show mpls forwarding** command displays whether an interface/port is either an Ethernet port or a POS port. For example, Ethernet port 3 on slot 2 is displayed as e2/3.

To display MPLS forwarding information, enter the **show mpls forwarding** command as shown in the example below.

NOTE
The output display from the **show mpls forwarding** command is applicable to both PowerConnect B-MLXe devices.

```
NetIron#show mpls forwarding
Total number of MPLS forwarding entries: 2
      Dest-prefix        In-lbl In-intf Out-lbl Out-intf Sig Next-hop
1     140.140.140.4/32                  3       e1/2     R   20.20.20.2
2     140.140.140.4/32                  1024    e1/10    R   20.20.20.2        DET
```

Syntax: show mpls forwarding [<*ip_prefix_addr*>/<*mask_len*> **[longer]** | <*ip_prefix_addr*> <*ip_mask*> **[longer]**]

TABLE 211 Output from the show mpls forwarding command

This field...	Displays...
dest-prefix	The destination FEC of the LSP.
in-lbl	The incoming segment or upstream label for the LSP. A value of 0 indicates the absence of the segment.
in-intf	The interface through which the label identified in the “in-lbl” column has been received for the LSP. A value of 0 indicates the absence of the segment. When applicable, the in-intf field also displays a VE interface specified by the <*vid*> variable.
out-lbl	The outgoing segment or downstream label for the LSP .
out-intf	The interface through which the label identified in the “out-lbl” column has been distributed for the LSP. The out-intf field displays whether an interface/port is an Ethernet port, POS port, or a VE interface. The VE interface ID specified by the <*vid*> variable. The out-intf display format for interface/port is as follows: • [e|p] <slot>/<port> “e” represents an Ethernet port “p” represents a POS port
sig	The signal protocol type associated with the label. Possible values are: • L – LDP • R – RSVP
next-hop	The next hop of the LSP.
DET	DET specifies that this is a Detour path.

Displaying RSVP information

You can display RSVP version information, the status of RSVP interfaces, RSVP session information, and RSVP statistics.

Displaying the RSVP version

To display the RSVP version number, as well as the refresh interval and refresh multiple.

```
NetIron #show mpls rsvp
Resource ReSerVation Protocol, version 1. rfc2205
RSVP protocol        = Enabled
R (refresh interval) = 30 seconds
K (refresh multiple) = 3
```

Syntax: show mpls rsvp

Displaying the status of RSVP interfaces

To display the status of RSVP on devices where it is enabled.

```
NetIron# show mpls rsvp interface
Interface State MD5 Auth
     e2/1 Up    OFF
     e2/2 Dn    OFF
     e4/1 Dn    OFF
     e4/2 Dn    OFF
```

Syntax: show mpls rsvp interface [brief]

NOTE
The **show mpls rsvp interface brief** command displays the same information as the **show mpls rsvp interface** command.

In this example, interfaces e 2/1, 2/2, 4/1, and 4/2 have been enabled for RSVP. Of these interfaces, interface e 2/1 can actively send and receive RSVP messages.

To display detailed information about RSVP-enabled interfaces.

```
NetIron# show mpls rsvp interface detail
Interface State MD5 Auth
     e2/1 Up    OFF
                        Total                  Since last clear
 PacketType        Sent      Received        Sent      Received
  Path             0         5745            0         5745
  Resv             5852      0               5852      0
  PathErr          0         0               0         0
  ResvErr          0         0               0         0
  PathTear         0         6               0         6
  ResvTear         0         0               0         0

 Errors                Total                 Since last clear
  Rcv MD5 Auth Errors  0                     0
```

Syntax: show mpls rsvp interface detail

For each RSVP-enabled interface, the following information is displayed

TABLE 212 Output from the **show mpls rsvp interface detail** command

This field...	Displays...
Interface	The RSVP-enabled interface can be an Ethernet interface or a VE-enabled interface. The VE interface ID is specified by the <*vid*> variable.
Path	The number of Path messages sent and received on the interface. Path messages store information about the state of the path along the LSRs in the LSP.
PathErr	The number of PathErr messages sent and received on the interface.
PathTear	The number of PathTear messages sent and received on the interface. PathTear messages cause path states to be deleted.
Resv	The number of Resv messages sent and received on the interface. Resv messages include FF (Fixed Filter), WF (Wildcard Filter), and SE (Shared Explicit) messages.
ResvErr	The number of ResvErr messages sent and received on the interface.
ResvTear	The number of ResvTear messages sent and received on the interface. ResvTear messages cause reservation states to be deleted.
MD5 Auth Errors	The number of MD5 authentication errors on received packets on this interface.

To clear the RSVP statistics counters, use the following command.

```
NetIron# clear mpls rsvp statistics
```

Syntax: clear mpls rsvp statistics

This command resets the counters listed under "Since last clear" for the **show mpls rsvp interface detail** and **show mpls rsvp statistics** commands.

Displaying RSVP session information

To display RSVP session information, use the following command.

```
NetIron(config)#show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress
Ingress RSVP:    10 session(s)
To              From                St Style Lbl_In  Lbl_Out Out_If LSPname
22.22.22.22     11.11.11.11         Up FF    -       3       e4/3 mlxe2
33.33.33.33     11.11.11.11(DI)     Up SE    -       3       e4/4   rj-vpls
33.33.33.33     11.11.11.11         Up SE    -       1039    e1/15  rj-vpls
..............
Transit RSVP:  1009 session(s)
To              From                St Style Lbl_In  Lbl_Out Out_If LSPname
22.22.22.22     33.33.33.33         Up SE    1024    3       e4/3   2
22.22.22.22     33.33.33.33(DI)     Up SE    1072    1319    e2/4   tomlxe2frr-
.............
Egress RSVP:     62 session(s)
To              From                St Style Lbl_In  Lbl_Out Out_If LSPname
11.11.11.11     22.22.22.22(DE)     Up SE    3       -       -      toxm1-frr
11.11.11.11     22.22.22.22(DE)     Up SE    3       -       -      toxm1-frr
11.11.11.11     22.22.22.22         Up SE    3       -       -      toxm1-frr
11.11.11.11     44.44.44.44         Up FF     3       -       -     tomlxe1
```

Syntax: show mpls rsvp session [name <*session-name*> **] [ingress | transit | egress] [brief | detail | extensive | wide | include]**

The **ingress** option limits the display to Ingress RSVP sessions.

The **transit** option limits the display to Transit RSVP sessions.

The **egress** option limits the display to Egress RSVP sessions.

The **brief** option limits the display to only brief RSVP session information.

The **detail** option displays detailed RSVP session information.

The **extensive** option displays extensive RSVP session information.

The **wide option** displays display the full LSP name in a single line.

NOTE
The **show mpls rsvp session brief** command displays the same information as the **show mpls rsvp session** command.

Table 213 describes the output of the **show mpls rsvp session** command.

TABLE 213 Output from the **show mpls rsvp session** command

This field...	Displays...
Ingress RSVP	Information about ingress RSVP sessions.
Transit RSVP	Information about transit RSVP sessions.
Egress RSVP	Information about egress RSVP sessions.
To	Destination (egress LER) of the session.
From	Source (ingress LER) of the session; the source address for the LSP that was configured with the **from** command.
St	State can be UP or DOWN.
Style	The RSVP reservation style. Possible values are FF (Fixed Filter), WF (Wildcard Filter), or SE (Shared Explicit).
Lbl_In	The label for inbound packets on this LSP.
Lbl_Out	The label applied to outbound packets on this LSP.
Out_If	The outbound interface displays the egress interface for a session. When applicable, the outbound interface displays a VE interface specified by the <*vid*> variable.
LSPname	The name of the LSP.

The **show mpls rsvp session detail** command displays if the session is downstream only, as shown in the following example.

NOTE
The output display from the **show mpls rsvp session detail** command is applicable to PowerConnect B-MLXe devices.

```
NetIron#show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress

Ingress RSVP:     1 session(s)
To              From                St Style Lbl_In  Lbl_Out Out_If LSPname
140.140.140.4   130.130.130.3(DI)   Up SE    -       1024    e1/10  to4
  Tunnel ID: 1, LSP ID: 1
  Time left in seconds (PATH refresh: 2, ttd: 4287904
                        RESV refresh: 26, ttd: 154)
  Tspec: peak 400000 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 2
   20.20.20.2 (S) -> 35.35.35.4 (S)
  Received RRO count: 2
   Protection codes/Rtr Id flag: P: Local  N: Node  B: Bandwidth  I: InUse R: RtrId
   20.20.20.2 -> 35.35.35.4
  Detour Sent: Number of PLR and Avoid Node ID pair(s): 1
   [1]: PLR: 30.30.30.3  Avoid Node: 0.0.0.0
  PATH rcvfrom: None (downstream only)
  PATH sentto:  20.20.20.2      (e1/10          ) (MD5 OFF)
  RESV rcvfrom: 20.20.20.2      (e1/10          ) (MD5 OFF)
```

Syntax: show mpls rsvp session detail

The **show mpls rsvp session** command with the **detail** option displays the same information described in Table 213, as well additional fields described in Table 214.

TABLE 214 Output from the **show mpls rsvp session detail** command

This field...	Displays...
Time left in seconds	The amount of time left for the PATH or RESV refreshes.
Tspec	Traffic engineering specification for the LSP, including the max-rate (“peak”), mean rate (“rate”), number of burst bytes (“size”), maximum policed unit (“M”—or maximum packet size), and minimum policed unit (“m”—or minimum packet size).
Explicit path hop count	The number of explicit hops used in this RSVP session.
Received RRO count	The number of Record Route Objects received on this RSVP session.
PATH sentto	Address of the next LSR in the LSP, and the interface used to reach this LSR. When applicable, PATH sentto displays a VE interface specified by the <*vid*> variable.
PATH rcvfrom	Address of the previous LSR in the LSP, and the interface used to reach this LSR. If the session is downstream only, then it will be displayed. When applicable, PATH rcvfrom displays a VE interface specified by the <*vid*> variable.

The **extensive** option provides the contents of the history buffer for the last 20 RSVP events, as shown in the following example.

```
NetIron#show mpls rsvp session extensive
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress
Ingress RSVP:     7 session(s)
To              From                St Style Lbl_In  Lbl_Out Out_If LSPname
33.33.33.33     11.11.11.11(DI)     Up SE    -       3       e4/4   rj-vpls
  Tunnel ID: 1, LSP ID: 1
  Time left in seconds (PATH refresh: 10, ttd: 4288020
                        RESV refresh: 0, ttd: 4288177)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 1
   129.0.0.6 (S)
  Received RRO count: 1
   Protection codes/Rtr Id flag: P: Local  N: Node  B: Bandwidth  I: InUse R: RtrId
   129.0.0.6
  Detour Sent: Number of PLR and Avoid Node ID pair(s): 1
   [1]: PLR: 41.1.1.1  Avoid Node: 41.1.1.2
  PATH sentto:  129.0.0.6       (e4/4          ) (MD5 OFF)
  RESV rcvfrom: 129.0.0.6       (e4/4          ) (MD5 OFF)
  PATH history:
    1 Dec 10 11:57:59 Query route to 33.33.33.33: nhop 129.0.0.6
    2 Dec 10 11:57:59 Tx PATH: out if(e4/4), flg(0x01000500/0x0000000a)
    3 Dec 10 11:57:59 Rx RESV: label(3), flg(0x01000500/0x0000000a)
    4 Dec 10 11:57:59 Tx cnnt req: hdl(0x0010c001), flg(0x01100500/0x0000000a)
    5 Dec 10 11:57:59 Start TC event(NEW_FLOW): action(0x0000000a)
    6 Dec 10 11:57:59 Rx cnnt resp: hdl(0x0010c001), flg(0x01100500/0x0000000a)
    7 Dec 10 11:57:59 Complete TC event(NEW_FLOW)
  RESV history:
    1 Dec 10 11:57:59 Add RSB: style(SE), filterSpec(1), flg(0x00000000)
    2 Dec 10 11:57:59 Add filterSpec: 11.11.11.11/1, label(3)
```

The **show mpls rsvp session** command with the **extensive** option displays the same information described in Table 213, Table 214, Table 216 and Table 217, as well as a history of the last 20 RSVP events with each event containing the following information:

- Event index (used to provide the total number of events).
- Time stamp.
- File name and line number where the event is logged.
- Event description and extra information associated with each event. For repeated events, such as route query, the attempt count indicates the number of times the event has occurred.

The **show mpls rsvp session wide** command allows the user to display the full LSP name in a single line. Previously, a long LSP name (greater than 12 characters) was text-wrapped in multiple lines. Now, the full LSP name can be displayed in a single line as shown in the following example.

NOTE
The **show mpls rsvp session wide** command is supported on PowerConnect B-MLXe devices.

```
NetIron#show mpls rsvp session wide
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress

Ingress RSVP:     4 session(s)
To              From                St Style Lbl_In  Lbl_Out Out_If LSPname
3.3.3.3         2.2.2.2             Up SE    -       3       e1/1   tunnel1
3.3.3.3         5.10.10.10(BI)      Dn -     -       -       e1/3   tunnel1
3.3.3.3         2.2.2.2(BYI)        Up SE    -       3       e1/3   by1
3.3.3.3         2.2.2.2             Up SE    -       3       e1/1   tunnelfromsanfranciscotonewyork
3.3.3.3         5.10.10.10(BI)      Dn -     -       -       e1/3   tunnelfromsanfranciscotonewyork
3.3.3.3         2.2.2.2(BYI)        Up SE    -       3       e1/3   bypasstunnelfromsfotonewyork

Transit RSVP:     0 session(s)
Egress RSVP:      0 session(s)
```

Syntax: show mpls rsvp session wide

The **include** option can be used with the **show mpls rsvp session wide** command to filter and display a specific RSVP session.

```
NetIron#show mpls rsvp session wide | include tunnelfromsanfranciscotonewyork
To              From                St Style Lbl_In  Lbl_Out Out_If LSPname
3.3.3.3         2.2.2.2             Up SE    -       3       e1/1   tunnelfromsanfranciscotonewyork
3.3.3.3         5.10.10.10(BI)      Dn -     -       -       e1/3   tunnelfromsanfranciscotonewyork
```

Syntax: show mpls rsvp session [**wide** [**| include** <*session-name*>]]

The <*session-name*> variable specifies the name of the RSVP session you want to display.

The **show mpls rsvp session wide** command can also be used with other RSVP session options, such as **backup**, **detour**, **ingress,** and so on to display the full LSP name in a single line. The following example displays the output from the **show mpls rsvp session backup wide** command.

NOTE
The **show mpls rsvp session wide backup** command and the **show mpls rsvp session backup wide** command can be used interchangeably. The output from both commands are the same.

```
NetIron#show mpls rsvp session backup wide
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress

Ingress RSVP:     2 session(s)
To              From                St Style Lbl_In  Lbl_Out Out_If LSPname
3.3.3.3         2.2.2.2             Up SE    -       3       e1/1   tunnel1
3.3.3.3         5.10.10.10(BI)      Dn -     -       -       e1/3   tunnel1
3.3.3.3         2.2.2.2             Up SE    -       3       e1/1   tunnelfromsanfranciscotonewyork
3.3.3.3         5.10.10.10(BI)      Dn -     -       -       e1/3   tunnelfromsanfranciscotonewyork

Transit RSVP:     0 session(s)
Egress RSVP:      0 session(s)
```

Syntax: show mpls rsvp session wide [backup | bypass |detour | ingress | egress | transit | name | up | down]

The **backup** option limits the display to backup RSVP sessions.

The **bypass** option limits the display to bypass RSVP sessions.

The **detour** option limits the display to detour RSVP sessions.

The **ingress** option limits the display to ingress RSVP sessions.

The **egress** option limits the display to egress RSVP sessions.

The **transit** option limits the display to transit RSVP sessions.

The **name** option limits the display to the RSVP session name.

The **up** option limits the display to an active RSVP session.

The **down** option limits the display to an inactive RSVP session.

Displaying RSVP statistics

The device constantly gathers RSVP statistics. RSVP statistics are collected from the time RSVP is enabled, as well as from the last time the RSVP statistics counters were cleared.

To display RSVP statistics.

```
NetIron# show mpls rsvp statistics
                      Total                   Since last clear
 PacketType        Sent      Received        Sent      Received
  Path             4         4               4         4
  Resv             4         4               4         4
  PathErr          0         0               0         0
  ResvErr          0         0               0         0
  PathTear         0         0               0         0
  ResvTear         0         0               0         0
  ResvConf         0         0               0         0

 Errors                 Total                Since last clear
  Rcv pkt bad length    0                    0
  Rcv pkt unknown type  0                    0
  Rcv pkt bad version   0                    0
  Rcv pkt bad cksum     0                    0
  Memory alloc fail     0                    0
  Rcv pkt processing error:
    Path                0                    0
    Resv                0                    0
    PathErr             0                    0
    ResvErr             0                    0
    PathTear            0                    0
    ResvTear            0                    0
    ResvConf            0                    0
```

Syntax: show mpls rsvp statistics

The following table describes the output of the **show mpls rsvp statistics** command.

TABLE 215 Output from the show mpls rsvp statistics command

This field...	Displays...
Path	The number of Path messages sent and received. Path messages store information about the state of the path along the LSRs in the LSP.
Resv	The number of RESV messages sent and received. RESV messages include FF (Fixed Filter), WF (Wildcard Filter), and SE (Shared Explicit) messages.
PathErr	The number of PathErr messages sent and received.

TABLE 215 Output from the show mpls rsvp statistics command (Continued)

This field...	Displays...
ResvErr	The number of ResvErr messages sent and received.
PathTear	The number of PathTear messages sent and received. PathTear messages cause path states to be deleted.
ResvTear	The number of ResvTear messages sent and received. ResvTear messages cause reservation states to be deleted.
ResvConf	The number of reservation confirmation messages sent and received.
Rcv pkt bad length	The number of times a packet was not processed because it was the wrong length.
Rcv pkt unknown type	The number of times an RSVP packet was not processed because it was not one of the types defined in RFC 2205.
Rcv pkt bad version	The number of times a packet was not processed because it was an RSVP version other than 1
Rcv pkt bad cksum	The number of times a packet was not processed because of a bad RSVP checksum.
Memory alloc fail	The number of times a packet was not processed because RSVP memory allocation failed on the device.
Rcv pkt processing error	
Path	The number of Path messages received with a packet processing error.
Resv	The number of RESV messages received with a packet processing error.
PathErr	The number of PathErr messages received with a packet processing error.
ResvErr	The number of ResvErr messages received with a packet processing error.
PathTear	The number of PathTear messages received with a packet processing error.
ResvTear	The number of reservation confirmation messages received with a packet processing error.
ResvConf	The number of reservation confirmation messages received with a packet processing error.

To clear the RSVP statistics counters.

```
NetIron# clear mpls rsvp statistics
```

Syntax: clear mpls rsvp statistics

This command resets the counters listed under “Since last clear” for the **show mpls rsvp interface detail** and **show mpls rsvp statistics** commands.

Displaying information about OSPF-TE LSAs

To display information about OSPF-TE LSAs.

```
NetIron# show ip ospf database link-state opaque-area

Area ID         Type    LS ID           Adv Rtr         Seq(Hex)   Age   Cksum
0               OpAr    1.0.0.0         3.3.3.3         80000006  1337  0x1a19
  Area-opaque TE LSA
  1 - router address (len 4): 3.3.3.3
Area ID      Type    LS ID           Adv Rtr         Seq(Hex)   Age      Cksum
0            OpAr    1.0.0.2         2.2.2.2         80000007  1333     0x88f1
  Area-opaque TE LSA
  2 - link (len 100):
    1 - link type (len 1): point-to-point(1)
    2 - link ID (len 4): 1.1.1.1
    3 - local i/f ip addr (len 4): 10.1.1.2
    4 - remote i/f ip addr (len 4): 10.1.1.1
    5 - TE metric (len 4):
    6 - max BW (len 4): 2372 Mbits/sec
    7 - max reservable BW (len 4): 2372 Mbits/sec
    8 - unreserved BW (len 32):
      Priority 0: 2372 Mbits/sec
      Priority 1: 2372 Mbits/sec
      Priority 2: 2372 Mbits/sec
      Priority 3: 2372 Mbits/sec
      Priority 4: 2372 Mbits/sec
      Priority 5: 2372 Mbits/sec
      Priority 6: 2372 Mbits/sec
      Priority 7: 2372 Mbits/sec
    9 - color (len 4): 0
```

Syntax: show ip ospf database link-state opaque-area

Displaying information about IS-IS LSPs with TE extensions

To display information about IS-IS LSPs with TE extensions.

```
NetIron# show isis database level2 detail

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL

MLXe3.00-00            0x00000644   0x78e3        843               1/0/0
   Area Address:  49.0002
   NLPID:  CC(IP)
   Hostname:  MLXe3
   Auth: Len 17 MD5 Digest "c33db90a87b93c80111980dbd59a19ed"
   TE  Router ID: 15.15.15.15
   Metric:  10       IP-Extended 15.15.15.15/32      Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 132.0.0.0/24        Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 121.0.0.0/24        Up: 0 Subtlv: 0
   Metric:  10       IS-Extended PE4.06
     Admin Group: 0x00000000
     Interface IP Address: 121.0.0.2
     Link BW: 1000000 kbits/sec
     Reservable BW: 1000000 kbits/sec
     Unreserved BW:
       [0]    1000000 kbits/sec  [1]    1000000 kbits/sec
       [2]    1000000 kbits/sec  [3]    1000000 kbits/sec
       [4]    1000000 kbits/sec  [5]    1000000 kbits/sec
       [6]    1000000 kbits/sec  [7]    1000000 kbits/sec
   Metric:  10       IS-Extended MLXe4.00
     Admin Group: 0x00000000
     Interface IP Address: 132.0.0.2
     Neighbor IP Address: 132.0.0.1
     Link BW: 10000000 kbits/sec
     Reservable BW: 10000000 kbits/sec
     Unreserved BW:
       [0]   10000000 kbits/sec  [1]   10000000 kbits/sec
       [2]   10000000 kbits/sec  [3]   10000000 kbits/sec
       [4]   10000000 kbits/sec  [5]   10000000 kbits/sec
       [6]   10000000 kbits/sec  [7]   10000000 kbits/sec
```

Syntax: show ip ospf database link-state opaque-area

Displaying MPLS Fast Reroute information

The following sections describe how to get information about MPLS Fast Reroute:

- "Displaying MPLS Fast Reroute LSP information"
- "Displaying RSVP session information"

Although the commands used to display MPLS and RSVP information are described in the PowerConnect *Configuration Guide*, new fields in the display present information for MPLS Fast Reroute. This section describes these new fields. For information about the other fields, see the equivalent display commands in this chapter.

Displaying MPLS Fast Reroute LSP information

To display MPLS Fast Reroute LSP information for a protected LSP that uses one-to-one (detour) backup.

```
NetIron# show mpls lsp frr_tunnel
LSP frr_tunnel, to 4.4.4.4
  From: 1.1.1.1, admin: UP, status: UP, tunnel interface: tnl4
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: p1, active: yes
  Path specific attributes:
   Tunnel interface: tnl4, outbound interface: e1/1
   Setup priority: 7, hold priority: 0
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   Tie breaking: random, hop limit: 0
   Explicit path hop counts: 3
    11.1.1.2 (S) -> 13.1.1.2 (S) -> 15.1.1.2 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    11.1.1.2 (PNB) -> 13.1.1.2 (PNB) -> 15.1.1.2
  Fast Reroute: one-to-one backup desired
   Bandwidth: 1024 kbps
   Detour LSP: UP, out-label: 1028, outbound interface: e1/3
```

Output that is shown in bold is unique to the **show mpls lsp** command when the LSP is configured for Fast Reroute by way of detour backup. The output is described in Table 216. Fields that are common to the output from the **show mpls lsp** command when an LSP is not configured for Fast Reroute are described in "Displaying signalled LSP status information" on page 1366.

TABLE 216 Output from the show mpls lsp command

This field...	Displays...
Fast Reroute	The method of Fast Reroute configured for this LSP. Currently only one-to-one backup is available.
Bandwidth	The bandwidth in Kilobits/sec for the bypass route. A value of 0 means that the detour route will use a best effort value for bandwidth.
Detour LSP	Indicates if the detour route is Up or Down.
out-label	The outbound label used when sending traffic over a detour LSP.
outbound interface	The physical interface on the router that is used for the detour route.

Displaying RSVP session information

To display RSVP Session information for a protected LSP.

NOTE
This section provides a brief example of the display from the **show mpls rsvp session** command. In the section titled "Example of MPLS Fast Reroute configuration" on page 1392, a detailed example of a network is given, and examples of output from the **show mpls rsvp session** command are displayed from each of the routers in the configuration, along with a descriptions of the relevant information in the displays.

```
NetIron# show mpls rsvp session name frr_tunnel
 Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
        DE:Egress Detour  RP:Repaired Session
 To              From                St    Style Lbl_in  Lbl_out  LSPname
 4.4.4.4         1.1.1.1(DI)         Up    SE    -       1028     frr_tunnel
   Time left in seconds (PATH refresh: 1, ttd: 4294621
                         RESV refresh: 20, ttd: 156)
   Tspec: peak 0 kbps rate 1024 kbps size 0 bytes m 20 M 65535
   Explicit path hop count: 3
    12.1.1.2 (S) -> 18.1.1.2 (S) -> 15.1.1.2 (S)
   Received RRO count: 3
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    12.1.1.2 -> 18.1.1.2 -> 15.1.1.2
   Detour Sent: Number of PLR and Avoid Node ID pair(s): 1
    [1]: PLR: 11.1.1.1  Avoid Node: 11.1.1.2
   PATH sentto:  12.1.1.2        (e1/3          ) (MD5 OFF)
   RESV rcvfrom: 12.1.1.2        (e1/3          ) (MD5 OFF)
  To              From                St    Style Lbl_in  Lbl_out  LSPname
 4.4.4.4         1.1.1.1             Up    SE    -       1029     frr_tunnel
   Time left in seconds (PATH refresh: 6, ttd: 146
                         RESV refresh: 20, ttd: 140)
   Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
   Fast Reroute: one-to-one backup desired
    Setup priority: 7, hold priority: 0
    Bandwidth: 1024 kbps, hop limit: 255
    Detour LSP: UP.  Nexthop (node) protection available.
     Up/Down times: 1, num retries: 0
   Explicit path hop count: 3
    11.1.1.2 (S) -> 13.1.1.2 (S) -> 15.1.1.2 (S)
   Received RRO count: 3
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    11.1.1.2 (PNB) -> 13.1.1.2 (PNB) -> 15.1.1.2
   PATH sentto:  11.1.1.2        (e1/1          ) (MD5 OFF)
   RESV rcvfrom: 11.1.1.2        (e1/1          ) (MD5 OFF)
```

For each RSVP-enabled interface, the following information is displayed

TABLE 217 Output from the show mpls rsvp interface detail command

This field...	Displays...
Explicit path hop count	The number of explicit hops used in this RSVP session.
Received RRO count	The number of Record Route Objects received on this RSVP session.
Detour Sent	Detour objects sent to the downstream node.
Detour Recvd	Detour objects received from the downstream node.
Number of PLR and Avoid Node ID pairs	Details of each detour object sent.

For each RSVP-enabled interface, the following information is displayed

TABLE 218 Output from the show mpls rsvp interface detail command

This field...	Displays...
DI	Ingress Detour. Specifies that this RSVP LSP is a detour LSP and it originates on this node (which is also a PLR of an FRR LSP).
DT	Transit Detour. Specifies that this RSVP LSP is a detour and is transiting this node.

TABLE 218 Output from the show mpls rsvp interface detail command (Continued)

This field...	Displays...
DM	Merged Detour. Specifies that this RSVP LSP is a detour LSP and it merges with another LSP (either a detour or a protected LSP) at this node (which is also a PLR for an FRR LSP).
DE	Egress Detour. Specifies that this RSVP LSP is a detour LSP and it is merging at an egress node of the protected path.
RP	Repaired Session. Specifies that this RSVP LSP is a protected LSP which has been locally repaired by the detour LSP on this node.

Syntax: show mpls rsvp session [brief | detail | *<session-name>* **}**

Using the **brief** option displays brief information about all of the RSVP sessions on the router.

Using the **detail** option displays detailed information about all of the RSVP sessions on the router.

If you specify a session name using the *<session-name>* variable, only the RSVP session specified is displayed in the detailed format.

Displaying MPLS configuration information

The **show mpls config** command lets you display all of the user-configured MPLS parameters. Using the **show mpls config** command, you can display all of the following global parameters configured on an PowerConnect router:

- ldp
- rsvp
- policy
- mpls-interface
- lsp
- path
- mpls vll
- mpls vpls
- vll local
- bypass-lsp

You can display the MPLS configuration information in any of the following modes brief, detail, and filters, as described in the sections that follow.

Displaying in the brief mode

In this mode, the information under router mpls policy, rsvp and ldp is displayed as shown in the following.

```
NetIron# show mpls config brief
router mpls
 policy
  admin-group m2 2
  traffic-eng isis level-1
  no propagate-ttl
  retry-limit 22
```

```
 rsvp
  refresh-interval 40

 ldp
  hello-timeout 12
  ka-interval 18
  advertise-labels for 5
  session 30.30.30.6 key 1 $!dZ@

end of MPLS configuration
```

Syntax: show mpls config brief

Displaying in the detail mode

In this mode, all of the MPLS global information and all of the MPLS configuration information are displayed as shown in the following.

```
NetIron# show mpls config
router mpls
 policy
  admin-group m2 2
  traffic-eng isis level-1
  no propagate-ttl
  retry-limit 22
 rsvp
  refresh-interval 40

 ldp
  hello-timeout 12
  ka-interval 18
  advertise-labels for 5
  session 30.30.30.6 key 1 $!dZ@

 mpls interfaces
  mpls-interface e1/1
  ldp-enable

  mpls-interface e1/2
  ldp-enable
  reservable-bandwidth 139000
  admin-group 2

 mpls paths
 path mu1_to_mu3
  strict 10.1.1.1
  strict 10.1.1.2
  strict 10.3.3.1
  strict 10.3.3.2
 path mu1_to_mu2_2
  strict 10.5.1.1
  strict 10.5.1.2
 path mu1_to_mu2_1
  strict 10.1.1.1
  strict 10.1.1.2
 lsp frr1
  to 10.4.2.1
  cos 6
  ipmtu 1028
```

```
 traffic-eng max-rate 180 mean-rate 125
 metric 5
 shortcuts ospf
 frr
   bandwidth 80
   hop-limit 55
 enable

lsp lsp13d
 to 10.3.3.2
 primary mu1_to_mu3
 cos 7
 traffic-eng max-rate 250 mean-rate 120
 no cspf
 enable

lsp lsp12d
 to 10.1.1.2
 cos 7
 traffic-eng max-rate 100 mean-rate 50
 enable

vll c13 5500
 vll-peer 33.33.33.1
 vlan 200
  tagged e 1/3

vll-local l15
 vlan 32
 untag e 1/4
  cos 4

vpls vpmaster 22
 vpls-peer 66.66.66.2
 vlan 110
 multicast active
 multicast pimsm-snooping

end of MPLS configuration
```

Syntax: show mpls config

Displaying MPLS configuration information for a VE interface

The **show mpls config** command and **show running-config** command display specific MPLS interface configuration information. If MPLS is configured on a VE interface, the VE interface ID is displayed in the output of the **show mpls config** command and the **show running-config** command.

The **show mpls config interface** command allows you to display configuration information for an MPLS-enabled interface. You can specify a VE interface on the CLI. The following example displays CLI commands executed for interface ve 20.

```
NetIron#show mpls config interface ve 20
  mpls-interface ve 20
  ldp-enable
```

Syntax: show mpls config interface [ethernet <*slot/port*> **| pos** <*slot/port*> **| ve** <*vid*>**]**

The **ve** parameter allows you to limit the display to VE interface ID specified by the <*vid*> variable.

Displaying filtered MPLS configuration information

An individual MPLS interface, LSP, VLL, bypass, or VPLS can be specified in the **show mpls config** command to display configuration of the specified object only. The following example displays the MPLS configuration information for the LSP named “frr1”.

```
NetIron# show mpls config lsp frr1
 lsp frr1
  to 10.4.2.1
  cos 6
  ipmtu 1028
  traffic-eng max-rate 180 mean-rate 125
  metric 5
  shortcuts ospf
  frr
    bandwidth 80
    hop-limit 55
  enable
```

Syntax: **show mpls config lsp** <*lsp-name*> | **path** <*path-name*> | **interface** <*interface-name*> | **vll** <*vll-name*> | **vll-local** <*vll-local-name*> | **vpls** <*vpls-name*> | **bypass** <*bypass-name*>

The **lsp** option lets you limit the display to configuration information for the LSP specified by <*lsp-name*>.

The **path** option lets you limit the display to configuration information for the path specified by <*path-name*>.

The **interface** option allows you to limit the display to configuration information for a specified MPLS interface. The <*interface-name*> can be either a POS , Ethernet, or VE -enabled interface. A POS or Ethernet interface is specified by interface type and slot/port. For example, ethernet 3/2 specifies an Ethernet interface on port 2 of the Interface module installed in slot 3. The VE interface ID specified by the <*vid*> variable.

The **vll** option lets you limit the display to configuration information for the VLL specified by <*vll-name*>.

The **vll-local** option lets you limit the display to configuration information for the Local VLL specified by the <*vll-local-name*> variable.

The **vpls** option lets you limit the display to configuration information for the VLL specified by <*vpls-name*>.

The **bypass-lsp** option lets you limit the display to information for the bypass LSP specified by <*bypass-name*>.

When an option is used without a variable specified, the configuration parameters for the option are shown for all elements that match the option are displayed. For instance, in the following example the **lsp** option is used without a specified <*lsp-name*> variable. Consequently, the display contains the configuration information for all three LSPs configured on the router.

```
NetIron# mpls config lsp
lsp frr1
  to 10.4.2.1
  cos 6
  ipmtu 1028
  traffic-eng max-rate 180 mean-rate 125
  metric 5
  shortcuts ospf
  frr
```

```
    bandwidth 80
    hop-limit 55
  enable

lsp lsp13d
  to 10.3.3.2
  primary mu1_to_mu3
  cos 7
  traffic-eng max-rate 250 mean-rate 120
  no cspf
  enable

lsp lsp12d
  to 10.1.1.2
  cos 7
  traffic-eng max-rate 100 mean-rate 50
  enable
```

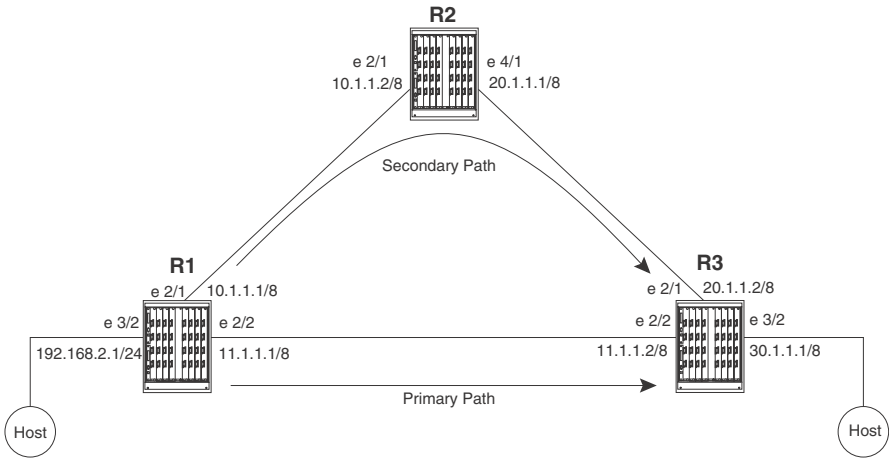
MPLS sample configurations

This section presents examples of typical MPLS configurations.

LSP with redundant paths

Figure 186 shows a signalled LSP configuration that has a primary and a secondary path. The destination for this LSP is 30.1.1.1. The primary path to this destination is through interface e 2/2, which has a direct link to interface e 2/2 on R3. If this link fails, the secondary path is established. The secondary path goes through R2.

FIGURE 186 LSP configuration with primary and secondary paths



Router R1 is the ingress LER for signalled LSP t3. Packets whose destination is 30.1.1.1 are assigned to this LSP. Two paths are configured, direct_conn and via_r2. Path direct_conn consists of a single strict node, 11.1.1.2, which is a directly connected interface on the destination LSR, R3.

Path via_r2 also consists of a single strict node, 10.1.1.2, a directly connected interface on R2. Since path via_r2 does not specify a node for R3, the hop between R2 and R3 is treated as a hop to a loose node. This means standard hop-by-hop routing is used to determine the path between R2 and R3.

Path direct_conn is the primary path for LSP t3, and path via_r2 is the secondary path. When the LSP is enabled, RSVP signalling messages set up path direct_conn. Packets assigned to this LSP use this path to reach the destination.

If path direct_conn fails, path via_r2 is set up, and packets assigned to LSP t3 then use path via_r2 to reach the destination. By default, the secondary path is not set up until the primary path fails. If you use the **standby** parameter in the configuration of the secondary path, both the primary and secondary paths are set up at the same time, although packets assigned to the LSP travel on the primary path only. If the primary path fails, the secondary path immediately carries the traffic.

Router R1

The following commands configure Router R1 in Figure 186.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface e 2/1 e 2/2
NetIron(config-mpls)# path direct_conn
NetIron(config-mpls-path)# strict 11.1.1.2
NetIron(config-mpls-path)# exit
NetIron(config-mpls)# path via_r2
NetIron(config-mpls-path)# strict 10.1.1.2
NetIron(config-mpls-path)# exit
NetIron(config-mpls)# lsp t3
NetIron(config-mpls-lsp)# to 30.1.1.1
NetIron(config-mpls-lsp)# primary direct
NetIron(config-mpls-lsp)# secondary via_r2
NetIron(config-mpls-lsp)# enable
NetIron(config-mpls-lsp)# exit
NetIron(config-mpls)# interface e 2/1
NetIron(config-e10000-2/1)# ip address 10.1.1.1 255.0.0.0
NetIron(config-e10000-2/1)# ip ospf area 1
NetIron(config-e10000-2/1)# exit
NetIron(config-mpls)# interface e 2/2
NetIron(config-e10000-2/2)# ip address 11.1.1.1 255.0.0.0
NetIron(config-e10000-2/2)# ip ospf area 1
NetIron(config-e10000-2/2)# exit
NetIron(config-mpls)# interface e 3/2
NetIron(config-if-e100-3/2)# ip address 192.168.2.1 255.255.255.0
NetIron(config-if-e100-3/2)# exit
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 1
NetIron(config-ospf-router)# exit
```

Router R2

In the configuration in Figure 186, Router R2 serves as a transit LSR for path via_r2. Since path via_r2 is the secondary path for the LSP, it will be used only if the primary path fails.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface e 2/1 e 4/1
```

```
NetIron(config-mpls)# interface e 2/1
NetIron(config-e10000-2/1)# ip address 10.1.1.2 255.0.0.0
NetIron(config-e10000-2/1)# ip ospf area 1
NetIron(config-e10000-2/1)# exit
NetIron(config-mpls)# interface e 4/1
NetIron(config-e10000-4/1)# ip address 20.1.1.1 255.0.0.0
NetIron(config-e10000-4/1)# ip ospf area 1
NetIron(config-e10000-4/1)# exit
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 1
NetIron(config-ospf-router)# exit
```

Router R3

In the configuration in Figure 186, Router R3 is the egress LER for LSP t3.

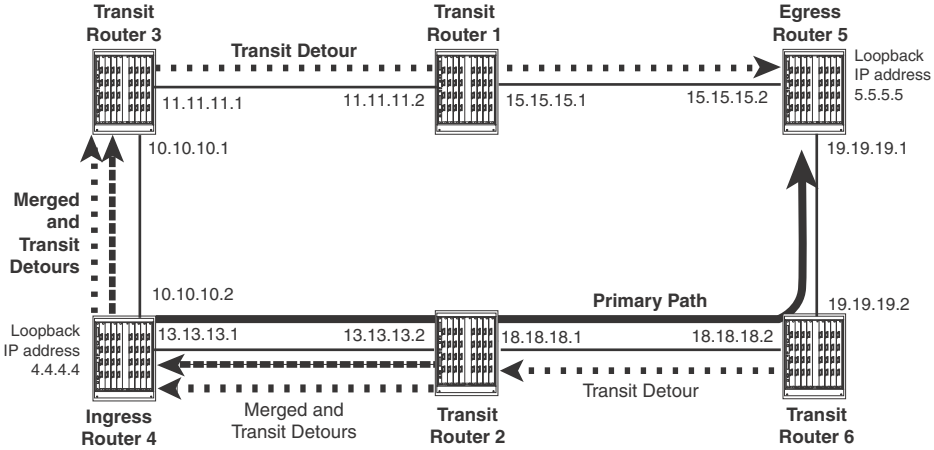
```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface e 2/1 e 2/2
NetIron(config-mpls)# interface pos 2/1
NetIron(config-e10000-2/1)# ip address 20.1.1.2 255.0.0.0
NetIron(config-e10000-2/1)# ip ospf area 1
NetIron(config-e10000-2/1)# exit
NetIron(config-mpls)# interface e 2/2
NetIron(config-e10000-2/2)# ip address 11.1.1.2 255.0.0.0
NetIron(config-e10000-2/2)# ip ospf area 1
NetIron(config-e10000-2/2)# exit
NetIron(config-mpls)# interface e 3/2
NetIron(config-if-e100-3/2)# ip address 30.1.1.1 255.0.0.0
NetIron(config-if-e100-3/2)# exit
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 1
NetIron(config-ospf-router)# exit
```

Example of MPLS Fast Reroute configuration

This example describes an MPLS Fast Reroute Loop Configuration. It provides the configuration required on Ingress Router 4 and examples of the **show mpls rsvp session** displays for all of the routers in the configuration. These examples show how the MPLS Fast Reroute configuration of an LSP affects the RSVP session on each of the routers in the configuration.

As illustrated in Figure 187 and described in the configuration example that follows, Ingress Router 4 is configured with a strict Label Switch Path A to Egress Router 5. In this configuration, if the path is broken between Ingress Router 4 and Egress Router 5, Transit Routers 2 or 6 will take a detour path back through Ingress Router 4 and continue through Transit Routers 3 and 1 to reach Egress Router 5.

FIGURE 187 MPLS Fast Reroute Loop configuration



The following is the MPLS Fast Reroute configuration for Ingress Router 4.

```
Router4(config)# interface loopback 1
Router4(config-lbif-1)# ip address 4.4.4.4/24
Router4(config)# interface ethernet 2/1
Router4(config-if-e1000-2/1)# ip address 10.10.10.2/24
Router4(config)# interface ethernet 2/9
Router4(config-if-e1000-2/9)# ip address 13.13.13.1/24
Router4(config)# router mpls
Router4(config-mpls)# mpls-interface ethernet 2/1 ethernet 2/9
Router4(config-mpls)# path a
Router4(config-mpls-path-a)# strict 2.2.2.2
Router4(config-mpls-path-a)# strict 6.6.6.6
Router4(config-mpls)# lsp 1
Router4(config-mpls-lsp-1)# to 5.5.5.5
Router4(config-mpls-lsp-1)# primary-path a
Router4(config-mpls-lsp-1)# frr
```

Displaying RSVP session information for example network

The **show mpls rsvp session** command, provides information regarding the primary and detour routes in an MPLS RSVP Fast Reroute enabled network. Display examples are provided for the following routers in the configuration shown in Figure 187:

- Transit Router 6
- Transit Router 2
- Ingress Router 4
- Transit Router 3
- Transit Router 1
- Egress Router 5

The following examples include displays for the **show mpls rsvp session** and **show mpls rsvp session detail** commands. For a general description of the command and its output refer to "Displaying RSVP session information" on page 1384.

The Transit Router 6 display

The following display examples are from Transit Router 6. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

Both displays show two paths from Ingress Router 4 at Loopback IP address 4.4.4.4. to Egress Router 5 at Loopback IP address 5.5.5.5. The (DI) path is an Ingress Detour path, and the path without a code is a protected path. The DI path is the detour path that is taken if Transit Router 6 is unable to use the primary path to Egress Router 5.

```
Router6# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DI)         Up    SE    1024    1025     1
5.5.5.5         4.4.4.4             Up    SE    1024    3        1

Egress RSVP:      0 session(s)
```

The following example displays the output from Transit Router 6 using the **show mpls rsvp session detail** command. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DI path, one PLR and Avoid Node ID pair is shown labeled [1]. In [1], the Point of Local Repair (PLR) is at IP Address 19.19.19.2 which is an interface on Transit Router 6 and the Avoid Node is IP address 0.0.0.0. The "Explicit path hop count" field indicates that there are five hops on this path from this router to the egress to the path at routers with the following IP addresses 18.18.18.1 (Transit Router 2), 13.13.13.1 (Ingress Router 4), 10.10.10.1 (Transit Router 3), 11.11.11.2 (Transit Router 1) and 15.15.15.2 (Egress Router 5) .

For the primary path, the "Explicit path hop count" field indicates that there is one hop on this path from this router to the egress to the path to the router at IP address 19.19.19.1 (Egress Router 5). The "Fast Reroute" field indicates that the primary path has been configured for one-to-one backup.

```
Router6# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DI)         Up    SE    1024    1025     1
  Time left in seconds (PATH refresh: 6, ttd: 4293497
                        RESV refresh: 24, ttd: 131)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Explicit path hop count: 5
   18.18.18.1 (S) -> 13.13.13.1 (S) -> 10.10.10.1 (S) -> 11.11.11.2 (S) ->
    15.15.15.2 (S)
  Received RRO count: 5
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   18.18.18.1 -> 13.13.13.1 -> 10.10.10.1 ->
    11.11.11.2 -> 15.15.15.2
```

```
  Detour Sent: Number of PLR and Avoid Node ID pair(s): 1
   [1]: PLR: 19.19.19.2  Avoid Node: 0.0.0.0
  PATH sentto:  18.18.18.1      (e5/1          ) (MD5 OFF)
  RESV rcvfrom: 18.18.18.1      (e5/1          ) (MD5 OFF)
To               From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5          4.4.4.4             Up    SE    1024    3        1
  Time left in seconds (PATH refresh: 28, ttd: 150
                        RESV refresh: 24, ttd: 128)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Fast Reroute: one-to-one backup desired
   Setup priority: 7, hold priority: 0
   Bandwidth: 0 kbps, hop limit: 255
   Detour LSP: UP.  Nexthop (node) protection available.
    Up/Down times: 1, num retries: 0
  Explicit path hop count: 1
   19.19.19.1 (S)
  Received RRO count: 1
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   19.19.19.1
  PATH rcvfrom: 18.18.18.1      (e5/1          ) (MD5 OFF)
  PATH sentto:  19.19.19.1      (e5/2          ) (MD5 OFF)
  RESV rcvfrom: 19.19.19.1      (e5/2          ) (MD5 OFF)
Egress RSVP:      0 session(s)
```

The Transit Router 2 display

The following display examples are from Transit Router 2. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

As for Transit Router 2, the displays show an Ingress Detour (DI) path, and a path without a code which identifies a protected path. In addition, a Merged Detour (DM) path is shown. The DM path is the detour path merged from Transit Router 6. All three paths are shown from Ingress Router 4 at Loopback IP address 4.4.4.4 to Egress Router 5 at Loopback IP address 5.5.5.5.

```
Router1# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To               From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5          4.4.4.4(DI)         Up    SE    1024    1024     1
5.5.5.5          4.4.4.4             Up    SE    1024    1024     1
5.5.5.5          4.4.4.4(DM)         Up    SE    1025    1024     1

Egress RSVP:      0 session(s)
```

The following example displays the output from Transit Router 2 using the **show mpls rsvp session detail** command. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DI path, two PLR and Avoid Node ID pairs are shown labeled [1] and [2]. In [1] the Point of Local Repair (PLR) is at IP Address 18.18.18.1 which is the interface to the primary path on Transit Router 2 and the Avoid Node is IP address 18.18.18.2 on Transit Router 6. In [2] the Point of Local Repair (PLR) is at IP Address 19.19.19.2 on Transit Router 6 and the Avoid Node is IP address 0.0.0.0. The "Explicit path hop count" field indicates that there are four hops on this path from this router to the egress of the path at routers with the following IP addresses 13.13.13.1 (Ingress Router 4), 10.10.10.1 (Transit Router 3), 11.11.11.2 (Transit Router 1) and 15.15.15.2 (Egress Router 5).

For the DM path, one PLR and Avoid Node ID pair is shown labeled [1]. In [1] the Point of Local Repair (PLR) is at IP Address 19.19.19.2 which is an interface on Transit Router 6 and the Avoid Node is IP address 0.0.0.0.

For the primary path, the “Explicit path hop count” field indicates that the path has two hops from this router to the egress to the path at routers with the following IP addresses 18.18.18.2 (Transit Router 6) and 19.19.19.1 (Egress Router 5). The “Fast Reroute” field indicates that the primary path has been configured for one-to-one backup.

```
Router2# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DI)         Up    SE    1024    1024     1
  Time left in seconds (PATH refresh: 1, ttd: 4293570
                        RESV refresh: 13, ttd: 141)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Explicit path hop count: 4
   13.13.13.1 (S) -> 10.10.10.1 (S) -> 11.11.11.2 (S) -> 15.15.15.2 (S)
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   13.13.13.1 -> 10.10.10.1 -> 11.11.11.2 ->
    15.15.15.2
  Detour Sent: Number of PLR and Avoid Node ID pair(s): 2
   [1]: PLR: 18.18.18.1  Avoid Node: 18.18.18.2
   [2]: PLR: 19.19.19.2  Avoid Node: 0.0.0.0
  PATH sentto:  13.13.13.1      (e5/10         ) (MD5 OFF)
  RESV rcvfrom: 13.13.13.1      (e5/10         ) (MD5 OFF)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4             Up    SE    1024    1024     1
  Time left in seconds (PATH refresh: 26, ttd: 151
                        RESV refresh: 13, ttd: 151)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Fast Reroute: one-to-one backup desired
   Setup priority: 7, hold priority: 0
   Bandwidth: 0 kbps, hop limit: 255
   Detour LSP: UP.  Nexthop (node) protection available.
    Up/Down times: 1, num retries: 0
  Explicit path hop count: 2
   18.18.18.2 (S) -> 19.19.19.1 (S)
  Received RRO count: 2
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   18.18.18.2 (PN) -> 19.19.19.1
  PATH rcvfrom: 13.13.13.1      (e5/10         ) (MD5 OFF)
  PATH sentto:  18.18.18.2      (e2/1          ) (MD5 OFF)
  RESV rcvfrom: 18.18.18.2      (e2/1          ) (MD5 OFF)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DM)         Up    SE    1025    1024     1
  Time left in seconds (PATH refresh: 31, ttd: 133
                        RESV refresh: 13, ttd: 141)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   13.13.13.1 -> 10.10.10.1 -> 11.11.11.2 ->
    15.15.15.2
```

```
  Detour Rcvd: Number of PLR and Avoid Node ID pair(s): 1
   [1]: PLR: 19.19.19.2  Avoid Node: 0.0.0.0
  PATH rcvfrom: 18.18.18.2      (e2/1          ) (MD5 OFF)
  RESV rcvfrom: 13.13.13.1      (e5/10         ) (MD5 OFF)
Egress RSVP:      0 session(s)
```

The Ingress Router 4 display

The following display examples are from Ingress Router 4. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

Like the display for Transit Router 2, the displays show an Ingress Detour (DI) path, a path without a code which identifies a protected path and Merged Detour (DM) path. The DM path is the detour path merged from Transit Routers 2 and 6. All three paths are shown from Ingress Router 4 at IP address Loopback 4.4.4.4 to Egress Router 5 at IP address Loopback 5.5.5.5. In the case of the DM path, a reroute at either Transit Router 2 or 6 will send traffic that had begun at Ingress Router 4 back through it, and forward through Transit Routers 3 and 1 to the ultimate destination at Egress Router 5.

```
Router4# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     1 session(s)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DI)         Up    SE    -       1028     1
5.5.5.5         4.4.4.4(DM)         Up    SE    1024    1028     1
5.5.5.5         4.4.4.4             Up    SE    -       1024     1

Transit RSVP:     0 session(s)
Egress RSVP:      0 session(s)
```

The following example displays the **show mpls rsvp session detail** output for Ingress Router 4. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DI path, three PLR and Avoid Node ID pairs are shown labeled [1], [2] and [3]. In [1] and [2] the Point of Local Repair (PLR) is at IP Address 13.13.13.1 which is the interface to the primary path on Ingress Router 4. The Avoid Node for pair [1] is IP address 13.13.13.2 on Transit Router 2 and the Avoid Node for pair [2] is IP address 18.18.18.2 on Transit Router 6. In [3] the Point of Local Repair (PLR) is at IP Address 18.18.18.1 which is the interface to the primary path on Ingress Router 2 and the Avoid Node for pair [3] is IP address 18.18.18.2 on Transit Router 6. The "Explicit path hop count" field indicates that there are three hops on the path from this router to the egress of the path at routers with the following IP addresses 10.10.10.1 (Transit Router 3), 11.11.11.2 (Transit Router 1) and 15.15.15.2 (Egress Router 5).

For the DM path, one PLR and Avoid Node ID pair is shown labeled [1]. In [1] the Point of Local Repair (PLR) is at IP Address 18.18.18.1 which is the interface to the primary path on Ingress Router 2 and the Avoid Node is IP address 18.18.18.2 on Transit Router 6.

For the primary path, the "Explicit path hop count" field indicates that there are three hops on this path from Ingress Router 4 to the egress to the path at routers with the following IP addresses 13.13.13.2 (Transit Router 2), 18.18.18.2 (Transit Router 6) and 19.19.19.1 (Egress Router 5). The "Fast Reroute" field indicates that the primary path has been configured for one-to-one backup.

```
Router4# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
```

```
Ingress RSVP:     1 session(s)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DI)         Up    SE    -       1028     1
  Time left in seconds (PATH refresh: 16, ttd: 4293608
                       RESV refresh: 27, ttd: 133)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 3
   10.10.10.1 (S) -> 11.11.11.2 (S) -> 15.15.15.2 (S)
  Received RRO count: 3
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.10.10.1 -> 11.11.11.2 -> 15.15.15.2
  Detour Sent: Number of PLR and Avoid Node ID pair(s): 3
   [1]: PLR: 13.13.13.1  Avoid Node: 13.13.13.2
   [2]: PLR: 13.13.13.1  Avoid Node: 18.18.18.2
   [3]: PLR: 18.18.18.1  Avoid Node: 18.18.18.2
  PATH sentto:  10.10.10.1      (e2/1          ) (MD5 OFF)
  RESV rcvfrom: 10.10.10.1      (e2/1          ) (MD5 OFF)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DM)         Up    SE    1024    1028     1
  Time left in seconds (PATH refresh: 6, ttd: 134
                       RESV refresh: 27, ttd: 133)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Received RRO count: 3
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.10.10.1 -> 11.11.11.2 -> 15.15.15.2
  Detour Rcvd: Number of PLR and Avoid Node ID pair(s): 1
   [1]: PLR: 18.18.18.1  Avoid Node: 18.18.18.2
  PATH rcvfrom: 13.13.13.2      (e2/20         ) (MD5 OFF)
  RESV rcvfrom: 10.10.10.1      (e2/1          ) (MD5 OFF)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4             Up    SE    -       1024     1
  Time left in seconds (PATH refresh: 37, ttd: 148
                       RESV refresh: 27, ttd: 152)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Fast Reroute: one-to-one backup desired
   Setup priority: 7, hold priority: 0
   Bandwidth: 0 kbps, hop limit: 255
   Detour LSP: UP.  Nexthop (node) protection available.
    Up/Down times: 1, num retries: 0
  Explicit path hop count: 3
   13.13.13.2 (S) -> 18.18.18.2 (S) -> 19.19.19.1 (S)
  Received RRO count: 3
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   13.13.13.2 (PN) -> 18.18.18.2 (PN) -> 19.19.19.1
  PATH sentto:  13.13.13.2      (e2/20         ) (MD5 OFF)
  RESV rcvfrom: 13.13.13.2      (e2/20         ) (MD5 OFF)
Transit RSVP:     0 session(s)
Egress RSVP:      0 session(s)
```

The Transit Router 3 display

The following display examples come from Transit Router 3. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

The display for Transit Router 3 shows a Transit Detour (DT) path. The DT path is the detour path that is an alternative to the primary path configured on Ingress Router 4 from itself to Egress Router 5. This detour path, shown from Ingress Router 4 at Loopback IP address 4.4.4.4 to Egress Router 5 at Loopback IP address 5.5.5.5, is only used if a link or router fails between the source and destination of the primary path.

```
Router3# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From             St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DT)         Up    SE    1028    1028     1
Egress RSVP:      0 session(s)
```

The following example displays the output from Transit Router 3 using the **show mpls rsvp session detail** command. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DT path, the "Explicit path hop count:" field shows that two hops exist from this router to the egress of the path at routers with the following at IP addresses: 11.11.11.2 (Transit Router 1) and 15.15.15.2 (Egress Router 5).

```
Router3# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5         4.4.4.4(DT)         Up    SE    1028    1028     1
  Time left in seconds (PATH refresh: 2, ttd: 141
                        RESV refresh: 25, ttd: 154)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Explicit path hop count: 2
   11.11.11.2 (S) -> 15.15.15.2 (S)
  Received RRO count: 2
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   11.11.11.2 -> 15.15.15.2
  PATH rcvfrom: 10.10.10.2      (e12/1         ) (MD5 OFF)
  PATH sentto:  11.11.11.2      (e11/18        ) (MD5 OFF)
  RESV rcvfrom: 11.11.11.2      (e11/18        ) (MD5 OFF)
Egress RSVP:      0 session(s)
```

The Transit Router 1 display

The following display examples are from Transit Router 1. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

The display for Transit Router 1 shows a Transit Detour (DT) path.The DT path is the detour path that is an alternative to the primary path configured on Ingress Router 4 from Ingress Router 4 to Egress Router 5. This detour path shown from Ingress Router 4 at Loopback IP address 4.4.4.4 to Egress Router 5 at Loopback IP address 5.5.5.5 is only used if there is a failed link or router between the source and destination of the primary path.

```
Router1# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From             St    Style Lbl_in Lbl_out LSPname
5.5.5.5         4.4.4.4(DT)         Up    SE    1028    3        1
Egress RSVP:      0 session(s)
```

The following example displays the output from Transit Router 1 using the **show mpls rsvp session detail** command. This option provides additional details about the paths described in the output from the **show mpls rsvp session** command

For the DT path, the "Explicit path hop count" field indicates that there is one hop from this router to the egress of the path at the router at IP address 15.15.15.2 (Egress Router 5).

```
Router3# show mpls rsvp session detail
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To               From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5          4.4.4.4(DT)         Up    SE    1028    3        1
  Time left in seconds (PATH refresh: 18, ttd: 146
                        RESV refresh: 15, ttd: 154)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Explicit path hop count: 1
   15.15.15.2 (S)
  Received RRO count: 1
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   15.15.15.2
  PATH rcvfrom: 11.11.11.1      (e7/16         ) (MD5 OFF)
  PATH sentto:  15.15.15.2      (e6/2          ) (MD5 OFF)
  RESV rcvfrom: 15.15.15.2      (e6/2          ) (MD5 OFF)
Egress RSVP:      0 session(s)
```

The Egress Router 5 display

The following display examples are from Egress Router 5. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

The display for Egress Router 5, shows an Egress Detour (DE) path and a path without a code that identifies a protected path. Both paths are shown from Ingress Router 4 at Loopback IP address 4.4.4.4 to Egress Router 5 at Loopback IP address 5.5.5.5. The primary path traverses Transit Routers 2 and 6. In the case of the DE path, a reroute sends traffic through Transit Routers 3 and 1.

```
Router4# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     0 session(s)
Egress RSVP:      1 session(s)
To               From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5          4.4.4.4(DE)         Up    SE    3       0        1
5.5.5.5          4.4.4.4             Up    SE    3       0        1
```

The following example displays the output from Egress Router 5 using the **show mpls rsvp session detail** command. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DE path, two PLR and Avoid Node ID pairs are shown labeled [1] and [2]. In both the Point of Local Repair (PLR) is at IP Address 13.13.13.1 which is the interface to the primary path on Ingress Router 4. The Avoid Node for pair [1] is IP address 13.13.13.2 on Transit Router 2 and the Avoid Node for pair [2] is IP address 18.18.18.2 on Transit Router 6.

The "Fast Reroute" field indicates that the primary path has been configured for one-to-one backup.

There is no "Explicit path hop count" field for either route because Egress Router 5 is the destination of the path.

```
Router5# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     0 session(s)
```

```
Egress RSVP:      1 session(s)
To               From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5          4.4.4.4(DE)         Up    SE    3       0        1
  Time left in seconds (PATH refresh: 18, ttd: 149
                       RESV refresh: 7, ttd: 152)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Detour Rcvd: Number of PLR and Avoid Node ID pair(s): 2
   [1]: PLR: 13.13.13.1  Avoid Node: 13.13.13.2
   [2]: PLR: 13.13.13.1  Avoid Node: 18.18.18.2
  PATH rcvfrom: 15.15.15.1      (e8/2          ) (MD5 OFF)
To               From                St    Style Lbl_in  Lbl_out  LSPname
5.5.5.5          4.4.4.4             Up    SE    3       0        1
  Time left in seconds (PATH refresh: 30, ttd: 152
                       RESV refresh: 7, ttd: 152)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Fast Reroute: one-to-one backup desired
   Setup priority: 7, hold priority: 0
   Bandwidth: 0 kbps, hop limit: 255
  PATH rcvfrom: 19.19.19.2      (e8/1) (MD5 OFF)
```

Examples of MPLS bypass LSP

This section contains **show** command output for protected LSPs and bypass LSPs. The section for bypass LSP shows information for a bypass configured on an Ethernet interface and a bypass configured on a LAG.

A **show mpls lsp** or **show mpls bypass-lsp** type of command must be entered at the ingress node of the LSP or bypass LSP, and the **show mpls rsvp session name** type of command should be used at transit nodes.

The subsections are separated into the following components:

- An interface with a bypass LSP protecting the interface, "Displaying an interface with bypass protection"
- Protected LSP configuration in an RSVP session, "Protected LSP shown in RSVP session"
- Bypass LSP shown in an RSVP session, "Bypass LSP in an RSVP session"
- An LSP that is requesting facility backup, "Displaying an LSP configured for bypass protection"
- Information when the bypass LSP is active, "A protected LSP while the bypass LSP is active"

Displaying an interface with bypass protection

This example shows that Ethernet interface 4/15 has one bypass LSP mlxe4-by. Bypass LSP mlxe4-by has, therefore, recorded at least this interface (and likely others) in its list of exclude interfaces. (Similarly, an interface can have multiple bypass LSPs protecting it. For example, the LSPs that traverse an interface might have destinations that make a single merge point impossible, so multiple bypass LSPs would be needed in this case to support different LSPs.).

```
NetIron#show mpls interface ethernet 4/15
e4/15
  Admin: Up  Oper: Up
  Maximum BW: 1000000 kbps, maximum reservable BW: 1000000 kbps
  Admin group: 0x00000000
  Reservable BW [priority] kbps:
    [0] 780000    [1] 780000    [2] 780000    [3] 760000
    [4] 760000    [5] 760000    [6] 760000    [7] 760000
  Last sent reservable BW [priority] kbps:
    [0] 780000    [1] 780000    [2] 780000    [3] 760000
    [4] 760000    [5] 760000    [6] 760000    [7] 760000
  Configured Protecting bypass lsps:
    MLXe4-by(UP)
```

Syntax: show mpls interface ethernet <*name*>

In the example that follows, interface e1/11 is on a LAG named Trunk3. One bypass LSP (mlxe2) is protecting the interface.

```
NetIron#show mpls interface
e1/11(Trunk3)
  Admin: Up  Oper: Up
  Maximum BW: 13000000 kbps, maximum reservable BW: 13000000 kbps
  Admin group: 0x00000000
  Reservable BW [priority] kbps:
    [0] 12981000    [1] 12981000    [2] 12981000    [3] 12981000
    [4] 12981000    [5] 12981000    [6] 12981000    [7] 12981000
  Last sent reservable BW [priority] kbps:
    [0] 12981000    [1] 12981000    [2] 12981000    [3] 12981000
    [4] 12981000    [5] 12981000    [6] 12981000    [7] 12981000
  Configured Protecting bypass lsps:
mlxe2(UP)
```

Syntax: show mpls interface <*name*>

Displaying bypass LSPs

This section has a variety of bypass LSP displays. The first example shows the running configuration. This output shows the name of the bypass LSP, its destination interface, the exclude interface e1/1 (of the protected LSP), and that the bypass LSP is enabled.

Syntax: show mpls bypass-lsp

To display any bypass LSPs that exist on the router, use the following command.

```
NetIron(config-if-e1000-2/15)#show mpls bypass-lsp
Note: LSPs marked with * are taking a Secondary Path
                              Admin Oper Tunnel   Up/Dn Retry Active
Name           To             State State Intf     Times No.   Path
mlxe1-2         11.11.11.11     UP    UP    tnl4     2     0     --
mlxe1-1         11.11.11.11     UP    UP    tnl3     2     0     --
mlxe4           44.44.44.44     UP    UP    tnl7     2     0 mlxe4-1
mlxe1           11.11.11.11     UP    UP    tnl2     2     0     --
mlxe1-5         11.11.11.11     UP    UP    tnl5     2     0     --
mlxe3           33.33.33.33     UP    UP    tnl6     2     0     --
```

Syntax: show mpls bypass-lsp

The following example displays details for the bypass LSP named mlxe1-2.

```
NetIron(config-if-e1000-2/15)#show mpls bypass-lsp mlxe1-2
LSP mlxe1-2, to 11.11.11.11
  From: 22.22.22.22, admin: UP, status: UP, tunnel interface (primary path): tnl4
  Times primary LSP goes up since enabled: 2
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: NONE, up: yes, active: yes
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 22000 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl4, outbound interface: e2/15
   Tunnel index: 5, Tunnel instance: 1 outbound label: 3
   Path calculated using constraint-based routing: yes
   Path calculated using interface constraint: yes
   Explicit path hop count: 1
    129.0.0.37 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    129.0.0.37
  exclude interface(s): e1/1
  Tunnel bandwidth
  Maximum BW: 22000 kbps
  Reservable BW [priority] kbps:
    [0] 22000    [1] 22000    [2] 22000    [3] 2000
    [4] 2000    [5] 2000    [6] 2000    [7] 2000
```

Syntax: show mpls bypass-lsp <*lsp_name*>

The <*lsp_name*> variable specifies the name of the LSP you want to display.

To display the detailed bypass LSP configuration under the router MPLS mode, enter the **show mpls bypass-lsp detail** command. The output from the **show mpls bypass-lsp detai**l command is enhanced to display a VE interface on the CLI as shown in the following example.

```
NetIron(config-mpls)#show mpls bypass-lsp detail
LSP to10_bp, to 128.128.128.28
  From: 125.125.125.1, admin: UP, status: UP, tunnel interface(primary
path): tnl2
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: NONE, up: yes, active: yes
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl2, outbound interface: ve 20
   Tunnel index: 3, Tunnel instance: 1 outbound label: 3
   Path calculated using constraint-based routing: yes
   Path calculated using interface constraint: no
   Explicit path hop count: 1
    90.90.90.28 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    90.90.90.28
  exclude interface(s): ve 54
```

Syntax: show mpls bypass-lsp detail

The **show mpls bypass-lsp wide** command allows the user to display the full bypass LSP name in a single line. Previously, a long LSP name (greater than 12 characters) was text-wrapped in multiple lines. Now, the full LSP name can be displayed in a single line as shown in the following example.

NOTE
The **show mpls bypass-lsp wide** command is supported on PowerConnect B-MLXe devices.

```
NetIron(config)#show mpls bypass-lsp wide
Note: LSPs marked with * are taking a Secondary Path
                                                        Admin Oper  Tunnel   Up/Dn Retry Active
Name                                                    To              State State Intf     Times No.   Path
by1                                                     3.3.3.3         UP    UP    tnl1     1     0     --
by2                                                     3.3.3.3         UP    UP    tnl2     1     0     --
bypasstunnelfromsanfranciscotonewyork                   3.3.3.3         UP    UP    tnl5     1     0
pathfromsanfrancisotonewyourk
```

Syntax: show mpls bypass-lsp wide

The **include** option can be used with the **show mpls bypass-lsp wide** command to filter and display specific bypass LSP name.

```
NetIron#show mpls bypass-lsp wide | include bypasstunnelfromsanfranciscotonewyork
                                                         Admin Oper Tunnel   Up/Dn Retry Active
Name                                                     To            State State Intf    Times No. Path
bypasstunnelfromsanfranciscotonewyork                    3.3.3.3         UP   UP   tnl5     1     0
pathfromsanfrancisotonewyourk
```

Syntax: show mpls bypass-lsp [**wide** [**| include** <*lsp_name*>]]

The <*lsp_name*> variable specifies the name of the LSP you want to display.

Bypass LSP in an RSVP session

Use the **show mpls rsvp session** command to display bypass LSP mlxe1-by. This example shows bypass LSP traversing a LAG, and the BYI field shows this is the bypass ingress.

```
NetIron#show mpls rsvp sess name mlxe1-by
Codes: DI:Ingress Detour DT:Transit Detour DM:Merged Detour
       DE:Egress Detour BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress
To              From                State Style Lbl_in  Lbl_out  LSPname
11.11.11.11     55.55.55.55(BYI)    Up    SE    -       1267 mlxe1-by
  Tunnel ID: 512, LSP ID: 1
  Time left in seconds (PATH refresh: 8, ttd: 148
                        RESV refresh: 10, ttd: 140)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 2
   123.0.0.13 (S) -> 129.0.0.9 (S)
  Received RRO count: 2
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   123.0.0.13 -> 129.0.0.9
  PATH sentto:  123.0.0.13     (p4/3(Trunk1)  ) (MD5 OFF)
  RESV rcvfrom: 123.0.0.13     (p4/3(Trunk1)  ) (MD5 OFF)
```

Syntax: show mpls rsvp sess name <*name*>

Bypass LSP in an RSVP session

Use the show RSVP session command to display bypass LSP mlxe1-by. This example shows bypass LSP traversing a LAG, and the BYI field shows this is the bypass ingress.

```
NetIron#show mpls rsvp sess name mlxe1-by
Codes: DI:Ingress Detour DT:Transit Detour DM:Merged Detour
       DE:Egress Detour BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress
To              From                St    Style Lbl_in  Lbl_out  LSPname
11.11.11.11     55.55.55.55(BYI)    Up    SE    -       1267 mlxe1-by
  Tunnel ID: 512, LSP ID: 1
  Time left in seconds (PATH refresh: 8, ttd: 148
                        RESV refresh: 10, ttd: 140)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 2
   123.0.0.13 (S) -> 129.0.0.9 (S)
  Received RRO count: 2
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   123.0.0.13 -> 129.0.0.9
  PATH sentto:  123.0.0.13     (p4/3(Trunk1)  ) (MD5 OFF)
  RESV rcvfrom: 123.0.0.13     (p4/3(Trunk1)  ) (MD5 OFF)
```

Syntax: show mpls rsvp session name <*name*>

Displaying an LSP configured for bypass protection

This example shows that:

- LSP mlxe3-120 is a candidate a bypass LSP. (The line “Fast Reroute facility backup desired” shows that LSP mlxe3-120 has requested facility backup.)
- The subsequent line shows that mlxe3-120 has selected bypass LSP mlxe1-by for protection.
- Bypass LSP mlxe1-by is up.
- The interface is on LAG p4/3.

```
NetIron#show mpls lsp mlxe3-120
LSP mlxe3-120, to 33.33.33.33
  From: 55.55.55.55, admin: UP, status: UP, tunnel interface(primary path): tnl35
  revert timer: 10 seconds
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: mlxe3-100, up: yes, active: yes
   Setup priority: 4, hold priority: 3
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   Tie breaking: random, hop limit: 0
  Sec. path: mlxe3-101, active: no
   Hot-standby: yes, status: up
   Setup priority: 7, hold priority: 0
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl35, outbound interface: e3/1
   Tunnel index: 36, Tunnel instance: 1 outbound label: 1032
   Path calculated using constraint-based routing: yes
   Path calculated using interface constraint: no
   Explicit path hop count: 4
    129.0.0.1 (S) -> 129.0.0.38 (S) -> 125.0.0.2 (S) -> 123.0.0.5 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    129.0.0.1 (PN) -> 129.0.0.38 (P) -> 125.0.0.2 ->
    123.0.0.5
  Fast Reroute: facility backup desired
   Backup LSP: UP, out-label: 1032, outbound interface: p4/3(Trunk1) bypass_lsp:
   FRR Forwarding State: Pri(active), Sec(up), Backup(up)
```

Syntax: show mpls lsp <*name*>

Protected LSP shown in RSVP session

Show the MPLS RSVP session for protected LSP mlxe3-199. The line “Backup LSP UP. Nexthop (node) protection available“ shows that protection is available for mlxe-199. If this LSP were actually riding the bypass LSP, this status would change from “protection available” to “in use.”.

```
NetIron-MLXe5(config-mpls-bypasslsp-123)#show mpls rsvp sess name mlxe3-199
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress
To               From                State Style Lbl_in  Lbl_out  LSP name
33.33.33.33      55.55.55.55         Up    SE    -       2399 mlxe3-199
  Tunnel ID: 121, LSP ID: 1
  Time left in seconds (PATH refresh: 11, ttd: 137
                        RESV refresh: 8, ttd: 138)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Fast Reroute: Facility backup desired
   Setup priority: 4, hold priority: 3
   Bandwidth: 0 kbps, hop limit: 255
   Backup LSP: UP.  Nexthop (node) protection available.
    Up/Down times: 1, num retries: 0
  Explicit path hop count: 4
   129.0.0.1 (S) -> 129.0.0.38 (S) -> 125.0.0.2 (S) -> 123.0.0.9 (S)
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   129.0.0.1 -> 129.0.0.38 -> 125.0.0.2 ->
    123.0.0.9
  PATH sentto:  129.0.0.1       (e3/1          ) (MD5 OFF)
  RESV rcvfrom: 129.0.0.1       (e3/1          ) (MD5 OFF)
To               From                State Style Lbl_in  Lbl_out  LSPname
33.33.33.33      129.0.0.2(BI)       Up    SE    -       3 mlxe3-199
  Tunnel ID: 121, LSP ID: 1
  Time left in seconds (PATH refresh: 0, ttd: 4196607
                        RESV refresh: 8, ttd: 4196765)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 1
   129.0.0.21 (S)
  Backup Sent
  PATH sentto: 129.0.0.21     (e2/13       ) (MD5 OFF)

  RESV rcvfrom: 129.0.0.21     (e2/13       ) (MD5 OFF)

  Riding bypass lsp: mlxe3-1
```

Syntax: show mpls rsvp session name <*name*>

A protected LSP while the bypass LSP is active

This section shows two views of a protected LSP while the bypass LSP is active.

To show a protected LSP while its bypass LSP is active, display the RSVP session for the LSP named mlxe3-120. This bypass LSP is on LAG p4/3, and two lines in the output show that mlxe3-120 is riding mlxe1-by.

```
NetIron@MLXe5#show mpls rsvp sess name mlxe3-120
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress
To              From                St    Style Lbl_in  Lbl_out  LSPname
33.33.33.33     55.55.55.55(RP)     Up    SE    -       1032 mlxe3-120
  Tunnel ID: 36, LSP ID: 1
  Time left in seconds (PATH refresh: 14, ttd: 141
                        RESV refresh: 36, ttd: 155)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Fast Reroute: Facility backup desired
   Setup priority: 4, hold priority: 3
   Bandwidth: 0 kbps, hop limit: 255
   Detour LSP: UP.  Nexthop (link) protection available and is in use.
    Up/Down times: 1, num retries: 0
  Explicit path hop count: 4
   129.0.0.1 (S) -> 129.0.0.38 (S) -> 125.0.0.2 (S) -> 123.0.0.5 (S)
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   129.0.0.9 -> 129.0.0.38 (P) -> 125.0.0.2 ->
    123.0.0.5
  RESV rcvfrom: 129.0.0.9       (p4/3(Trunk1)  ) (MD5 OFF)
 Riding bypass lsp: mlxe1-by

To              From                St    Style Lbl_in  Lbl_out  LSP name
33.33.33.33     129.0.0.2(BI)       Up    SE    -       1032 mlxe3-120
  Tunnel ID: 36, LSP ID: 1
  Time left in seconds (PATH refresh: 37, ttd: 155
                        RESV refresh: 36, ttd: 155)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 4
   129.0.0.9 (S) -> 129.0.0.38 (S) -> 125.0.0.2 (S) -> 123.0.0.5 (S)
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   129.0.0.9 -> 129.0.0.38 (P) -> 125.0.0.2 ->
    123.0.0.5
  Backup Sent
  PATH sentto:  129.0.0.9       (p4/3(Trunk1)  ) (MD5 OFF)
  RESV rcvfrom: 129.0.0.9       (p4/3(Trunk1)  ) (MD5 OFF)
  Riding bypass lsp: mlxe1-by
```

Syntax: show mpls rsvp session name <*name*>

The following command shows an LSP that is using its bypass. Note the last lines of output.

```
NetIron@MLXe5#show mpls lsp mlxe3-120
LSP mlxe3-120, to 33.33.33.33
  From: 55.55.55.55, admin: UP, status: UP, tunnel interface(primary path): tnl35
  revert timer: 10 seconds
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: mlxe3-100, up: yes (backup), active: yes
   Setup priority: 4, hold priority: 3
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   Tie breaking: random, hop limit: 0
  Sec. path: mlxe3-101, active: no
   Hot-standby: yes, status: up
   Setup priority: 7, hold priority: 0
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl35, outbound interface: p4/3(Trunk1)
   Tunnel index: 36, Tunnel instance: 1 outbound label: 1032
   Path calculated using constraint-based routing: yes
   Path calculated using interface constraint: no
   Explicit path hop count: 4
    129.0.0.9 (S) -> 129.0.0.38 (S) -> 125.0.0.2 (S) -> 123.0.0.5 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    129.0.0.9 -> 129.0.0.38 (P) -> 125.0.0.2 ->
    123.0.0.5
  Fast Reroute: facility backup desired
   Backup LSP: UP, out-label: 1032, outbound interface: p4/3(Trunk1) bypass_lsp:
   FRR Forwarding State: Pri(down), Sec(up), Backup(active)
```

Syntax: show mpls lsp <*name*>

Configuring Label Distribution Protocol (LDP) 31

LDP overview

The following list displays the Label Distribution Protocol (LDP) features supported by PowerConnect B-MLXe:

- LDP
- LDP ECMP for transit LSR
- LDP Hello Interval and Hold Timeout Values
- LDP Message Authentication
- New encryption code for passwords, authentication keys, and community strings
- Option of FEC Type for Auto-discovered VPLS Peers
- MPLS Signalling: LDP support
- Resetting LDP neighbor
- LDP over RSVP (for transit LSR only)
- Displaying the LDP version
- Displaying Information about Specified LDP-Enabled Interfaces
- Displaying LDP FEC information
- Displaying information for a specified LDP FEC type
- Displaying LDP FEC summary information
- Displaying LDP FEC VC information
- Displaying information for a specified LDP FEC VC
- Displaying LDP Neighbor Connection Information
- Displaying the LDP Packet Statistics

The devices support Label Distribution Protocol (LDP) for setting up non-traffic-engineered tunnel LSPs in an MPLS network. LDP is described in RFC 3036.

When used to create tunnel LSPs, LDP allows a set of destination IP prefixes (known as a Forwarding Equivalence Class or FEC) to be associated with an LSP. Each LSR establishes a peer relationship with its neighboring LDP-enabled routers and exchanges label mapping information. This label mapping information is stored in an LDP database on each LSR. When an LSR determines that one of its peers is the next hop for a FEC, it uses the label mapping information from the peer to set up an LSP that is associated with the FEC. It then sends label mapping information to its upstream peers, allowing the LSP to extend across the MPLS network.

The devices advertise their loopback addresses to their LDP peers as a 32-bit “prefix” type FEC. When an LSR installs a label for a FEC, it also creates an MPLS tunnel route, which is then made available to routing applications. This allows each router to potentially be an ingress LER for an LSP whose destination is the device's loopback address.

The result of an LDP configuration is a full mesh of LSPs in an MPLS network, with each LDP-enabled router a potential ingress, transit, or egress LSR, depending on the destination.

The implementation supports the following aspects of LDP

Liberal label retention – Each LSR sends its peers Label Mapping messages, which map a label to a FEC. Peer LSR receiving these messages retain all of the mappings, even though they may not actually be used for data forwarding.

Unsolicited label advertisement – The LSR sends Label Mapping messages to its LDP peers even though they did not explicitly request them.

Ordered label distribution – The LSR sends a Label Mapping message to its peers only when it knows the next hop for a FEC, or is itself an egress LER for the FEC. If an LSR does not know the next hop for a FEC, and is not an egress LER for the FEC, it waits until a downstream LSR sends it a Label Mapping message for the FEC. At this point, the LSR can send Label Mapping messages for the FEC to its peers. This allows label mappings to be distributed, in an orderly fashion, starting from the egress LER and progressing upstream.

The Multi-Service IronWare software the LDP label space ID has a default value of zero which improves interoperability with routers from other venders. Also, to provide backward compatibility with Multi-Service IronWare software previous versions, a command lets you change the LDP label space ID value to 1 as described in "Resetting LDP neighbors" on page 1421.

Configuring LDP on an interface

To use LDP, a loopback address (with a 32-bit mask) ***must*** be configured on the LSR. The first loopback address configured on the device is used in its LDP identifier. If the loopback address used in the LDP identifier is removed, all LDP functions on the LSR will be shut down. LDP sessions between the LSR and its peers will be terminated, and LDP-created tunnels will be removed. If other loopback interfaces are configured on the device, the lowest-numbered loopback address will then be used as a new LDP identifier. LDP sessions and tunnels will be set up using this new LDP identifier.

To configure LDP on an interface, enter commands such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface e 1/2
NetIron(config-mpls)# ldp-enable
```

Syntax: [no] ldp-enable

NOTE
You should enable LDP on the same set of interfaces that IGP routing protocols such as OSPF and IS-IS are enabled.

Configuring an option of FEC type for auto-discovered VPLS peers

By default, Dell uses FEC 129 to send the VC label binding for auto-discovered VPLS peers. There are mixed environments where VPLS static configured peers and auto-discovered peers exist. In these environments, the following VPLS command allows you to configure FEC 128 for all VPLS. Enter the command such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# fec-128-for-auto-disc-peers
```

Syntax: [no] fec-128-for-auto-disc-peers

The default value is FEC 129.

Using the **no** option returns a configuration that was previously changed from the default value back to the default value.

NOTE
You must reload your system for this command to take effect.

LDP ECMP for transit LSR

NOTE
LDP ECMP for transit LSR is supported only on PowerConnect B-MLXe devices.

LDP Equal-Cost Multi-Path (ECMP) for transit LSR provides ECMP support for transit routers on an LDP LSP. The LDP LSP tunnel at ingress will continue to be a single ECMP path.

ECMP programming for LDP transit LSP creates a set of ECMP paths on the forwarding plane at any transit router. LDP LSPs transit traffic is load balanced using programmed ECMP. The number of ECMP paths that are used depends on the number of eligible paths that are available, and the maximum number of LDP ECMP paths configured by the user. The number of available paths sent to LDP are controlled by the Routing Table Manager (RTM) which is limited by IP load sharing. LDP will also enable its own load sharing limit. The lesser of the two load sharing limits will form the maximum number of ECMP paths that can be programmed on forwarding plane. For more information on configuring the maximum number of LDP ECMP paths, refer to "Changing the maximum number of LDP ECMP paths" on page 1414.

When new ECMP paths are added, or existing paths are deleted from a set of eligible ECMP paths, MPLS forwarding decides if these changes will lead to a different set of paths to be used for LDP LSP, ingress tunnel, or transit LSP. If a different set of paths is used, updates are sent to the forwarding plane. MPLS only sends an update to the forwarding plane if there is a change to the set of programmed paths. MPLS always sends the complete set of ECMP paths to the forwarding plane. If the user changes the load sharing configuration, updates are also sent to the forwarding plane. FEC updates are only generated if the new load sharing value is different from the set of ECMP paths programmed in the forwarding plane.

NOTE
LDP ECMP is not supported at the ingress router.

The ingress LDP LSP can be different from the transit LSP for the same FEC. If all ECMP paths provided by the RTM are using LDP tunneling enabled for RSVP shortcut LSP(s), then the ingress LDP tunnel is not created.

Changing the maximum number of LDP ECMP paths

To change the maximum number of LDP ECMP paths, enter the following commands under the router MPLS mode.

```
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# load-sharing 4
```

Syntax: load-sharing <*num*>

The <*num*> variable specifies the maximum number of LDP ECMP paths. Enter a number from 1 through 8. The default value is 1. To return to the default maximum number of LDP ECMP paths, use the **no** form of the command.

The load sharing configuration is displayed in the output of the **show mpls ldp** command.

NOTE
The configuration for load sharing is displayed only if the configured value is different from the default value.

```
NetIron#show mpls ldp
Label Distribution Protocol version 1
LSR ID: 125.125.125.1, using Loopback 1 (deleting it will stop LDP)
Hello interval: Link 5 sec, Targeted 15 sec
Hold time value sent in Hellos: Link 15 sec, Targeted 45 sec
Keepalive interval: 6 sec, Hold time multiple: 6 intervals
load-sharing 4
```

Syntax: show mpls ldp

The load sharing configuration can also be displayed in the output of the **show mpls config** command, as shown in the following example.

```
NetIron#show mpls config
router mpls
 policy
  no propagate-ttl
 ldp
  load-sharing 4
```

Syntax: show mpls config

MPLS OAM support for LDP ECMP

MPLS OAM support for traceroute at any transit router returns the list of labels used at that transit router. However, traceroute is not able to exercise all ECMP paths. The forwarding plane selects one ECMP path to forward OAM packets. All traversed labels that were returned at each transit router are displayed at the NetIron router originating the traceroute.

Display changes to commands for LDP ECMP

The output from the **show mpls forwarding** command has been updated to display all ECMP paths programmed in the forwarding plane.

NOTE
The In-lbl field will not display a value for ingress LDP LSP.

```
NetIron#show mpls forwarding
Total number of MPLS forwarding entries: 7
   Dest-prefix     In-lbl  In-intf Out-lbl Out-intf   Sig  Next-hop
 1 21.21.21.21/32                    3        e2/3    L    80.80.80.1
 2 21.21.21.21/32  1026              3        e2/3    L    80.80.80.1
 3 11.11.11.11/32                  1028       ve4     L    90.90.90.25
 4 11.11.11.11/32  1029            1028       ve4     L    90.90.90.25
 5 11.11.11.11/32  1029              3        tnnl1   L    11.11.11.11
 6 11.11.11.11/32  1029              3        tnnl2   L    11.11.11.11
 7 11.11.11.11/32  1029              3        tnnl3   L    11.11.11.11
```

Syntax: show mpls forwarding

The output from the **show mpls ldp fec prefix** command has been updated to display all available ECMP mappings on outgoing interfaces, and their corresponding next hops. When the outgoing interface is an RSVP tunnel, the RSVP LSP name is displayed as the outgoing interface in the out_if field. The following example displays multiple next hops created in LDP from ECMP.

```
NetIron#show mpls ldp fec prefix 11.11.11.11
FEC_CB: 0x362eee00, idx: 14, type: 2, pend_notif: None
 State: current, Ingr: Yes, Egr: No, UM Dist. done: Yes
 Prefix: 11.11.11.11/32
 next_hop: 90.90.90.25, out_if: ve4
 next_hop: 11.11.11.11, out_if: tunnelto12_3
 next_hop: 11.11.11.11, out_if: tunnelto12
 next_hop: 11.11.11.11, out_if: tunnelto12_2

 Downstream mappings:
 Local LDP ID        Peer LDP ID         Label     State     CB
 128.128.128.28:0    11.11.11.11:0       3         Installed 0x34db96c4(-1)
 128.128.128.28:0    125.125.125.1:0     1028      Installed 0x34db9e64(-1)

 Upstream mappings:
 Local LDP ID        Peer LDP ID         Label     CB
 128.128.128.28:0    11.11.11.11:0       1029      0x34db98ac(-1)
 128.128.128.28:0    21.21.21.21:0       1029      0x34db97b8(-1)
```

Syntax: show mpls ldp fec prefix <*IPaddress*>

The output from the **show mpls ldp path** command has been updated to display all available ECMP paths for a specified FEC. If LDP selects an RSVP tunnel as its outgoing interface, the RSVP tunnel name is displayed in the ingress interface (intf) field.

The order of fields has changed in the **show mpls ldp path** command output. The Destination route field is displayed first, followed by the Upstr-session (label) field, and the Downstr-session (label,intf) field. The Upstr-session field and the Downstr-session field display an entry for an upstream session, and a downstream session if there is an entry to be displayed. If the outgoing interface is a tunnel, the ingress interface (intf) field displays the tunnel name instead of the LSP name, as shown in the following example.

```
NetIron#show mpls ldp path
Destination route Upstr-session(label)  Downstr-session(label,intf)
11.11.11.11/24    11.11.11.11:0 (1029)  90.90.90.25:0  (1028 ve4)
                  21.21.21.21:0 (1029)  11.11.11.11:0  (3 tnnl1)
                                        11.11.11.11:0  (3 tnnl2)
                                        11.11.11.11:0  (3 tnnl3)
```

Syntax: show mpls ldp path

The **show mpls statistics label** command has been updated to display statistics for LDP ECMP paths. In the following example, packet count for all ECMP paths are collected.

```
NetIron(config-mpls)#show mpls statistics label 2/1
In-label    In-Port(s)     In-Packet Count
1024        e2/1 - e2/20       100
```

Syntax: show mpls statistics label

Setting the LDP Hello Interval and Hold Timeout values

The LDP Hello interval and Hello Hold Timeout timers are used to establish Hello Adjacency between peers. The Hello Interval is the time period between which the LSR sends out Hello messages and the Hello Hold Timeout value is the amount of time that the sending LSR maintains its record of Hellos from the receiving LSR without receipt of another Hello message.

The Hello interval and Hello Hold Timeout timer values can be obtained from the global default values, configured globally on a router, or in the case of the Hello Hold Timeout timer configured per-interface. When configuring these values the following constraints must be followed:

- the Hello Interval value must be < 32767
- the Hello Hold Timeout value must be <65535
- The Hello Hold Timeout value must be >= 2 * Hello Interval value

As described in the following sections, values can be set that determine the values used on the configured router and values sent to adjacent peers for their configuration:

- Setting the LDP Hello Interval and Hold Timeout Values
- Setting the LDP Hold Time Sent to Adjacent LSRs
- Determining the LDP Hold Time on an MPLS Interface

Setting the LDP Hello interval values

The LDP hello interval controls how often the device sends out LDP Hello messages. Hello messages are used to maintain LDP sessions between the device and its LDP peers. You can set the interval for LDP Link Hello messages (LDP Hello messages multicast to all routers on the sub-net), as well as for LDP Targeted Hello messages (LDP Hello messages unicast to a specific address, such as a VLL peer):

- **For targeted LDP sessions –** the LDP Hello Interval can only be set globally. This configuration is described in "Setting the LDP Hello Interval globally for targeted LDP sessions" on page 1417. If a Hello Interval is not set for Targeted LDP sessions, then the global default value is used.
- **For link LDP sessions –** the LDP Hello Interval can be set globally which applies to all LDP interfaces or on a per-interface basis. The LDP Hello Interval values in Link LDP sessions are determined by the following procedure in the order described below.

1. If the Hello Interval is set per-interface, that value is used. This configuration is described in "Setting the LDP Hello Interval per-Interface (link Only)" on page 1418.
2. If the Hello Interval is not set per-interface, then the value set for LDPs globally is used. This configuration is described in "Setting the LDP Hello Interval globally for link LDP sessions" on page 1417.
3. If the Hello Interval is not set either globally or per-interface, the global default value is used.

If Hello Adjacency already exists, the adjacency remains up and any new configured interval takes effect upon the expiration of the current Hello Interval timer. Consequently, the next and subsequent hello messages are sent at the new interval.

Setting the LDP Hello Interval globally

You can set a global LDP Hello Interval that applies to all LDP sessions, regardless of interface. This is performed separately for Link and Targeted LDP sessions as described in the following sections:

- Setting the LDP Hello Interval Globally for Link LDP Sessions
- Setting the LDP Hello Interval Globally for Targeted LDP Sessions

Setting the LDP Hello Interval globally for link LDP sessions

To set the interval for LDP Link Hello messages to 10 seconds, enter the following command.

```
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# hello-interval 10
```

Syntax: [no] hello-interval <*seconds*>

The <*seconds*> variable specifies the value in seconds of the Hello Interval that you are globally configuring for LDP Link Hello messages. The LDP hello interval can be from 1 – 32767 seconds. The default value for LDP Link Hello messages is 5 seconds.

The value set here can be overridden on a per-interface basis as described in "Setting the LDP Hello Interval per-Interface (link Only)" on page 1418.

The **no** option removes a previously configured LDP Link Hello Interval.

Setting the LDP Hello Interval globally for targeted LDP sessions

To set the interval for LDP Targeted Hello messages to 20 seconds, enter the following command.

```
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# hello-interval target 20
```

Syntax: [no] hello-interval target <*seconds*>

The <*seconds*> variable specifies the value in seconds of the Hello Interval that you are globally configuring for LDP Targeted messages. The LDP hello interval can be from 1 – 32767 seconds. When you set a new LDP hello interval, it takes effect immediately. The default value for LDP Targeted Hello messages is 15 seconds.

The **no** option removes a previously configured LDP Targeted Hello Interval.

NOTE
This value can only be set globally for all Targeted LDP sessions on the router. Per-interface configuration is only available for Link LDP sessions.

Setting the LDP Hello Interval per-Interface (link Only)

You can set the LDP Hello Interval on a per-Interface basis. This option is only available for Link LDP sessions. The following example configures the MPLS Interface at Ethernet port 1/1 with a **hello-interval** of 10 seconds.

```
NetIron(config)# mpls
NetIron(config-mpls)# mpls-interface ethernet 1/1
NetIron(config-mpls-if-e100-1/1)# ldp-params
NetIron(config-mpls-if-e100-1/1-ldp-params)# hello interval 10
```

Syntax: [no] hello-interval <*seconds*>

The <*seconds*> variable specifies the value in seconds of the Hello Interval that you are configuring on this MPLS interface for LDP Link Hello messages.

No default value exists for this parameter. If a value is set here, it overrides any LDP Hello Interval that was globally configured. However, if no value is set for this parameter, it defaults either to the LDP Hello Interval that was configured globally or, if no value was configured globally, to the default global value. For information about the global configuration, refer to "Setting the LDP Hello Interval globally for link LDP sessions" on page 1417.

The **no** option removes a previously configured LDP Hello Interval.

Setting the LDP hold time sent to adjacent LSRs

The LDP hold time specifies how long the device waits for its LDP peers to send a Hello message. If the device does not receive a Hello message within this time, the LDP session with the peer can be terminated. The device includes the hold time in the Hello messages it sends out to its LDP peers.

The LDP Hold Time sent in Hello messages to adjacent LSRs can be configured globally for either Link or Targeted LDP sessions, as described in the following sections:

- Setting the LDP Hello Hold Time Sent to Adjacent LSRs for Link LDP Sessions
- Setting the LDP Hello Hold Time Sent to Adjacent LSRs for Targeted LDP Sessions

Setting the LDP Hello hold time sent to adjacent LSRs for link LDP sessions

To set the hold time included in LDP Link Hello messages to 20 seconds, enter the following command.

```
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# hello-timeout 20
```

Syntax: [no] hello-timeout <*seconds*>

The <*seconds*> variable specifies the value in seconds of the LDP hello timeout that is sent in Hello messages to Link LDP peers. The range for this value is 1 – 65535 seconds. The default value is 15 seconds.

When you globally set a LDP hold time, the new time takes effect immediately and goes in the next Hello message sent. This hold time applies to only the hold time that the device sends to its peers; it does not affect the hold time the device uses to time out those peers. The latter is determined from the hold time that peers send to the device.

The **no** option removes a previously configured LDP Hello Timeout value and returns the value to the default.

Setting the LDP Hello hold time sent to adjacent LSRs for Targeted LDP sessions

To set the hold time included in LDP Targeted Hello messages to 60 seconds, enter the following command.

```
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# hello-timeout target 60
```

Syntax: [no] hello-timeout target <*seconds*>

The <*seconds*> variable specifies the value in seconds of the LDP hello timeout that is sent in Hello messages to Targeted LDP peers. The LDP hold time can be from 1 – 65535 seconds. The default value is 45 seconds.

The **no** option removes the previous LDP Hello Timeout Target value and returns the value to the default.

Determining the LDP Hold Time on an MPLS interface

An MPLS interface uses the LDP Hello Hold Time to determine how long it waits for its LDP peers to send a Hello message. How this determination is made differs for a targeted LDP session and a link LDP session, as follows:

- **For targeted LDP sessions** – The value received in Hello messages from its peers determines the time that the device waits for its LDP peers to send a Hello message. If the Timeout value received from a peer is zero, the Hold Time is set to the default period of 45 seconds.
- **For link LDP sessions** – In this case, the wait time is determined by any one of the below criteria.

1. If the Hello Hold Time is set per-interface, that value is used. That value is set as described in "Setting the LDP Hello Holdtime per-interface (link only)" on page 1420.
2. If the Hello Hold Time is not set per-interface, the hold time in the received message is used.
3. If the Hello Hold Time in the received message is 0, the default value of 15 seconds is used.

Setting the LDP Hello Holdtime per-interface (link only)

You can set the LDP Hello Holdtime on a per-interface basis. This holdtime value is sent in Hello messages from the interface. This option is available for Link LDP sessions only. The following example configuration is for the MPLS Interface at Ethernet port 1/3 with a **hello-timeout** of 18 seconds.

```
NetIron(config)# mpls
NetIron(config-mpls)# mpls-interface ethernet 1/3
NetIron(config-mpls-if-e100-1/3)# ldp-params
NetIron(config-mpls-if-e100-1/3-ldp-params)# hello-timeout 18
```

Syntax: [no] hello-timeout <*seconds*>

The value configured in the <*seconds*> variable is the LDP Hello Timeout value that will be sent in LDP Hello messages from this interface. The minimum value that can be configured for this variable is 2 * the value set for the Hello Interval.

The **no** option removes a previously configured LDP Hello Timeout value and sets the value as described in "Determining the LDP Hold Time on an MPLS interface" on page 1419.

LDP message authentication

The Multi-Service IronWare software supports LDP authentication based upon the TCP MD5 signature option specified in RFC 2385. This RFC defines a new TCP option for carrying an MD5 digest in a TCP segment. The purpose of this feature is to protect against spoofed TCP segments in a connection stream.

Configuring LDP message authentication

The PowerConnect Series routers allow configuration of an authentication key on a per LDP session basis. The LDP session can be to an adjacent peer (basic discovery) or to the targeted peer (extended discovery). This feature must be configured on both sides of an LDP peer link. To configure LDP message authentication use the following commands.

```
NetIron(config)# mpls
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# session 10.10.10.3 key early
```

Syntax: [no] session <*remote-ip-addr*> **key** <*string*>

The <*remote-ip-addr*> variable specifies the IP address of the LDP peer that authentication is being configured for.

The <*string*> variable specifies a text string of up to 80 characters used for authentication between LDP peers. It must be configured on both peers.

By default, **key** is encrypted. If you want the authentication key to be in clear text, insert a **0** between **key** and <*string*>.

Example

```
NetIron(config-mpls-ldp)# session 10.10.10.3 key 0 early
```

The software adds a prefix to the key string in the configuration. For example, the following portion of the code has the encrypted code "2".

```
session 1.1.1.1 key 2 $XkBTb24tb0RuXA==
```

The encrypted code can be one of the following:

- 0 = the key string is not encrypted and is in clear text
- 1 = the key string uses proprietary simple crytographic 2-way algorithm
- 2 = the key string uses proprietary base64 crytographic 2-way algorithm

Resetting LDP neighbors

This feature allows you to reset or clear an MPLS LDP neighbor session. The session will be terminated and re-established (if at least one LDP "hello" adjacency exists with the peer). As a result of LDP session termination, the following database associated with the LDP session will also be cleared. Once the session is re-established, these session specific database will be re-learned from it's peer:

- LDP downstream and upstream label database ("**show mpls ldp database ...**")
- LDP label switched path ("**show mpls ldp path ...**")
- LDP peer ("**show mpls ldp peer ...**")
- LDP created MPLS tunnels ("**show mpls ldp tunnel ...**")
- LDP FECs learned from the resetting neighbor sessions (**show mpls ldp fec ...**"). FECs are actually not cleared immediately, but it is marked that no LDP session exists.

To reset/clear an MPLS LDP neighbor session

Syntax: clear mpls ldp neighbor [all | *<peer-ip-addr>* **[label-space-id** *<label-space>***]**

If the **all** option is specified, all LDP sessions on the router will be reset, including the targeted LDP sessions.

An LDP session is uniquely referred to by *<peer-ip-addr>* **:** *<label-space>*. This command also allows you to input *<peer-ip-addr>* only and ignore *<label-space>*. In this case, all LDP sessions with the matching peer address will be reset.

Executing this command displays a warning message if the LDP session is not found corresponding to the supplied *<peer-ip-addr>* (and *<label-space>*). If an LDP session is not in operational state, resetting it will have no impact.

Resetting LDP neighbor considerations

The **clear mpls ldp neighbor** feature terminates the specified LDP sessions. The LDP sessions are automatically reestablished if at least one "hello" adjacency exists with the neighbor, and LDP configuration remains unchanged. This command allows a user to reset the following LDP sessions:

- Platform-wide label space
- Interface specific label space

When an LDP session is terminated as a result of the **clear mpls ldp neighbor** command, the switch will not generate any notification message for the neighbor. Instead, the device will unilaterally terminate the session and close the associated TCP session. The other end of the LDP session will detect this reset operation in either of the following two ways:

- TCP session is broken (half connected). The device will detect this while receiving or sending LDP messages on TCP socket fails (with fatal error), indicating that underlying TCP session is aborted by remote peer.

- Receives a new TCP connection request from the neighbor while the older session is still operational (if this is in passive role)

NOTE
Either of above events will trigger the remote end of the LDP session to tear down the session and try to reestablish. Resetting an LDP session impacts the associated VPLS/VLL sessions. Resetting an LDP session which is not in operational state will have no impact.

Validating LDP session reset

You can check the following LDP session specific parameters to validate that a session has been successfully reset:

- The LDP session state will transition from "Operational" to "Nonexistant" upon clearing it. It may quickly transition from “Nonexistant" to "Operational.” In that case, the **show mpls ldp session** [detail | A.B.C.D] will show the "Up time", and that must have been reset to zero upon clearing the session.
- The LDP session specific database (mentioned above) is cleaned upon resetting the LDP session
- The TCP port number (on the active end of the LDP session) may have been changed once the LDP session comes up after reset. In other words, the TCP port number before the reset and after the reset may be different. Use the command **show mpls ldp session** [detail | A.B.C.D] to view the TCP port number.
- Syslog will log the event of a LDP session going down and then coming back up, as a result of resetting the LDP session. Use the command **show log** to view the syslog events.

Following is an example of how to use the **show log** command to view the syslog.

```
NetIron# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 33 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning


Dynamic Log Buffer (50 lines):
Sep  9 18:38:20:N:MPLS: LDP entity session 1.1.1.1:0 with peer 2.2.2.2:0 is up
Sep  9 18:38:02:N:MPLS: LDP entity session 1.1.1.1:0 with peer 2.2.2.2:0 is dow
```

The following command shows two LDP sessions with neighbor 31.234.123.64.

```
NetIron#show mpls ldp session
Peer LDP ID             State          Adj Used  My Role   Max Hold  Time Left
31.234.123.64           Operational    Link      Passive   36        33
```

The following command clears both the link and targeted LDP session with neighbor 31.234.123.64, because the <*label_space*> optional parameter has not been specified.

```
NetIron# clear mpls ldp neighbor 31.234.123.64
NetIron#
NetIron#show mpls ldp session
Peer LDP ID             State          My Role   Max Hold  Time Left
31.234.123.64           Operational    Passive   36        33
NetIron#
```

This command shows that after waiting for roughly 20 seconds (depends on the hello/keepalive timer periodicity), both the LDP sessions are reestablished.

```
NetIron# clear mpls ldp neighbor 31.234.123.64
Peer LDP ID              State          My Role   Max Hold  Time Left
31.234.123.64            Operational    Passive   36        33
```

You can also validate the **clear mpls ldp** `neighbor` command using the syslog command.

```
NetIron# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 47 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
Sep  9 19:23:24:N:MPLS: LDP entity session 2.2.2.2:0 with peer 31.234.123.64 is
up
Sep  9 19:23:08:N:MPLS: LDP entity session 2.2.2.2:10 with peer 31.234.123.64 is
down
Sep  9 19:23:08:N:MPLS: LDP entity session 2.2.2.2:0 with peer 31.234.123.64 is
down
```

LDP over RSVP (for transit LSR only)

LDP over RSVP (for transit LSR only) enables LDP traffic to tunnel across RSVP tunnels. The RSVP tunnel is the transit of the LDP tunnel. On PowerConnect B-MLXe, LDP over RSVP can run over all types of LSPs (for example, one-to-one or facility Fast ReRoute (FRR) LSPs, adaptive LSPs, or redundant LSPs).

NOTE
LDP over RSVP configuration (for transit LSR only) is supported on PowerConnect B-MLXe devices.

LDP over RSVP is supported for all cases except when a NetIron router acts as a Label Edge Router (LER) for both LDP and RSVP. On the transit for LDP, the RSVP tunnel (RSVP LSP with LDP tunneling enabled) is used to reach the next-hop. The RSVP tunnel is treated as a single hop, and thus external LDP FECs are not advertised to the LSRs which are part of the RSVP core.

Currently, LDP depends on the Routing Table Manager (RTM) to provide the best next-hop for a particular prefix when LDP decides which label (received from its downstream peers) should be installed. This does not change for LDP over RSVP configuration. For LDP to install a label received from a non-directly connected peer whose route is through an RSVP tunnel, LDP must receive the corresponding route from the RTM indicating that the RSVP tunnel is used to reach the next-hop.

LDP over RSVP is supported under the following conditions:

- The RTM provides MPLS with a shortcut route for a particular prefix.
- The shortcut route must be an IS-IS, OSPF, or BGP shortcut.
- The RSVP tunnel must be enabled for LDP tunneling. For more information on enabling LDP tunneling, refer to "Enabling LDP over RSVP" on page 1424.

NOTE
If an RSVP tunnel is created on ingress LSR with IS-IS or OSPF shortcuts enabled, and LDP tunneling is also enabled, then the LDP tunnel to the egress router of the RSVP tunnel is not formed. An LDP tunnel is not created at the ingress LSR if RTM selects the RSVP tunnel as the next-hop to the destination.

Previously, prefix FECs were not advertised to a targeted peer, and prefix FECs received from a targeted peer were not installed. However with LDP over RSVP support, if a targeted session is used for LDP over RSVP, prefix FECs are advertised to its targeted peer, and prefix FEC received from a targeted peer is installed.

A targeted LDP session is brought up if any one of the following configurations exist:

- A targeted peer address is set up on the egress router of an RSVP tunnel. For more information on configuring a targeted peer address, refer to "Configuring a targeted peer address" on page 1426.
- The user enables RSVP LSP with LDP tunneling configured. For more information on configuring LDP tunneling, refer to "Enabling LDP over RSVP" on page 1424.
- A Layer 2 VPN (VLL or VPLS) peer is configured.

Enabling LDP over RSVP

To enable LDP traffic to tunnel across an RSVP tunnel, first create an LSP, then enable LDP tunneling on the LSP as shown in the following example.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp blue
NetIron(config-mpls-lsp-blue)#to 20.20.20.20
NetIron(config-mpls-lsp-blue)# ldp-tunneling
```

The following message appears on the CLI.

```
This LSP can be used for LDP tunneling if it is used as a shortcut
```

This message implies that if an RSVP tunnel is used as an ISIS or OSPF shortcut, then the shortcut must be explicitly configured.

NOTE
There is no configuration needed for BGP shortcut.

To enable ISIS shortcuts or OSPF shortcuts, enter the **shortcuts isis** command, or the **shortcuts ospf** command as shown in the following example.

```
NetIron MLX-4 Router(config-mpls-lsp-blue)#shortcuts isis level2
NetIron MLX-4 Router(config-mpls-lsp-blue)#enable
Connecting signaled LSP blue
```

Syntax: [no] ldp-tunneling

Syntax: [no] shortcuts isis level1 | level 2

Syntax: [no] shortcuts ospf

By default, LDP tunneling is disabled. The user must disable the LSP configuration to change the setting on the **ldp-tunneling** command.

NOTE
The **ldp-tunneling** command is not available under bypass LSP configuration.

To disable IS-IS shortcuts or OSPF shortcuts, enter the **no** form of the command.

The **level1** or **level2** keyword is required and indicates the level of IS-IS routing enabled on the device. The levels are:

- level1 – A level1 router routes traffic only within the area that includes the router. To forward traffic to another area, a level1 router sends the traffic to the nearest level2 router.
- level2 – A level2 router routes traffic between areas within a domain.

The LDP tunneling configuration is displayed in the output of the **show mpls lsp** command. If LDP tunneling is enabled, the line reads “yes.” If it is not enabled, the line reads “no.”

```
NetIron(config-mpls)#show mpls lsp blue
LSP blue, to 20.20.20.20
  From: 10.10.10.10, admin: UP, status: DOWN (Path not sent)
  Times primary LSP goes up since enabled: 0
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 1
  Pri. path: NONE, up: no, active: no
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  LDP tunneling enabled: yes
```

Syntax: show mpls lsp <*lsp_name*>

The <*lsp_name*> variable specifies the LSP name you want to display.

The LDP tunneling configuration is also displayed in the output of the **show mpls config** command, and the **show mpls config lsp** command. In the following example, LDP tunneling with ISIS shortcuts is enabled.

```
NetIron MLX-4 Router(config-mpls)#show mpls config lsp blue
 lsp blue
  to 20.20.20.20
  shortcuts isis level2
  ldp-tunneling
  enable
```

Syntax: show mpls config lsp <*lsp_name*>

The <*lsp_name*> variable specifies the LSP name you want to display.

The output from the **show mpls config** command displays a list of configured peer addresses as shown in the following example.

```
NetIron#show mpls config
router mpls
 policy
  traffic-eng isis level-2
  ingress-tunnel-accounting
 ldp
  session 7.7.7.2 key 2 $LSFVPW9iIQ==
  session 7.7.7.3 key 2 $LSFVPW9iIQ==

 bfd
  min-tx 50 min-rx 50 multiplier 3
mpls-interface e3/3
  ldp-enable
  admin-group 2

  mpls-interface e3/17
  rsvp-authentication 2 key $LSFVPW9iIQ==
  ldp-enable
```

Syntax: show mpls config

Configuring a targeted peer address

Currently, the NetIron router does not send a targeted Hello message in response to receiving a targeted Hello message from a peer that is not configured as a L2VPN peer. In the case when a L2VPN peer is not configured on a NetIron router, and the user would like to enable support for LDP over RSVP, the user must specify the IP address of the peer to bring up a targeted session. To trigger a targeted session that is set up on the egress router of an RSVP tunnel, enter the **targeted-peer** command under the MPLS LDP configuration as shown in the following example.

```
NetIron(config)# router mpls
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# targeted-peer 10.10.10.10
```

Syntax: [no] targeted-peer <*ip-address*>

The <*ip-address*> variable specifies the IP address of the targeted peer. To disable the configuration, enter the **no** form of the command.

NOTE
A targeted peer address is not configured on the ingress router of the RSVP tunnel because configuring LDP tunneling for the LSP automatically brings up a targeted session.

Displaying targeted peer addresses

To display a list of configured peer addresses, enter the **show mpls ldp targeted-peer** command on the CLI as shown in the following example.

```
NetIron# show mpls ldp targeted-peer
Peer_address
2.2.2.2
```

Syntax: show mpls ldp targeted-peer

TTL propagation for LDP over RSVP packets

TTL propagation for LDP over RSVP packets is controlled by the **propagate-ttl** command, and the **label-propagate-ttl** command:

- If the label operation involves the swap of the LDP label followed by the push of the RSVP label, the **label-propagate-ttl** command will control the propagation of the LDP label TTL to the RSVP label TTL. By default, the TTL is not propagated. The RSVP label TTL is set to 255. If the **label-propagate-ttl** command is configured by the user, the LDP label TTL is propagated to the RSVP label TTL.
- If the label operation involves the pop (or the removal) of the LDP label followed by the push of the RSVP label, the following two cases are considered:
  - If the LDP label is not the only label in the Layer 2 or Layer 3 VPN stack, the **label-propagate-ttl** command will control the propagation of TTL from the outer LDP label to the VC label and, in turn, from the VC label to the RSVP label. By default, the **label-propagate-ttl** command is turned off. The VC label TTL is not affected when the LDP tunnel label and the RSVP label TTL are set to 255.
  - If the LDP label is the only label in the IP over MPLS stack, the **propagate-ttl** command controls the propagation of TTL from the LDP label to the IP header and, in turn, from the IP header to the RSVP label. By default, the **propagate-ttl** command is turned on.

By default, TTL propagation is enabled for IP over MPLS traffic when an RSVP and an LDP tunnel terminate on the same node. For traceroute purposes, if an RSVP tunnel is traced, then TTL propagation should be enabled.

NOTE
For consistent behavior in all cases of TTL propagation for LDP over RSVP packets, Dell recommends that the user always turn on or turn off both the **label-propagate-ttl** command and the **propagate-ttl** command.

Enabling TTL propagation

By default, MPLS traceroute will not display the LSRs the RSVP tunnel is transiting through, except when the egress router is acting as the egress for both the LDP and the RSVP tunnel. In other words, the RSVP tunnel is treated as a single hop. The **label-propagate-ttl** command and the **propagate-ttl** command must be enabled in order to display details of the RSVP core. By default, the **propagate-ttl** command is enabled. To trace an RSVP path, enable the **label-propagate-ttl** command on all NetIron routers along the RSVP path, as shown in the following example.

```
NetIron(config)# router mpls
NetIron#(config-mpls)# policy
NetIron#(config-mpls-policy)# label-propagate-ttl
```

Syntax: [no] label–propagate-ttl

To disable the configuration, enter the **no** form of the command. By default, the **label-propagate-ttl** command is turned off. When MPLS traceroute is configured through an RSVP core, FEC validation for LDP FEC is not preformed at the transit LSR of the RSVP tunnel.

Class of Service (CoS) treatment for LDP over RSVP

The following sections describe COS treatment for LDP over RSVP (transit) for ingress RSVP and RSVP Penultimate Hop Pop (PHP).

Ingress RSVP

The internal priority of the ingress RSVP is mapped from the incoming LDP label EXP bits. If the RSVP tunnel has a COS configured, it will override the internal priority of the ingress RSVP. By default, the EXP bits in the outgoing RSVP label are mapped from the internal priority. If the **qos exp encoding off** command is configured on the outgoing interface, the RSVP label EXP bits are set to the internal priority of the ingress RSVP. The incoming LDP label EXP bits are preserved in the outgoing LDP label EXP bits irrespective of whether the **qos exp encoding** command is turned on or off on the outgoing interface.

RSVP PHP

The internal priority is mapped from the incoming RSVP label EXP bits. On the egress router of the RSVP tunnel, the outgoing LDP label EXP bits are set to the incoming LDP label EXP bits. This is irrespective of whether the **qos exp encoding** command is turned on or off on the outgoing interface.

Displaying LDP information

You can display the following information about LDP:

- The LDP version number, as well as the LSP's LDP identifier and loopback number
- Information about active LDP-created LSPs on the device
- Information about LDP-created tunnel LSPs for which this device is the ingress LER
- The contents of the LDP database
- Information about the LDP session between this LSR and its LDP peers
- Information about the connection between this LSR and its LDP peers
- Information about LDP-enabled Interfaces on the LSR

Displaying the LDP version

NOTE
The **show mpls ldp** command has changed. The Num VC FEC currently allocated field is no longer displayed in the output of the **show mpls ldp** command.

To display the LDP version number, the LSR ID and loopback number, and the LDP hello interval and hold time, enter the **show mpls ldp** command shown in the example below. T

```
NetIron(config)# show mpls ldp
Label Distribution Protocol version 1
  LSR ID: 2.2.2.2, using Loopback 1 (deleting it will stop LDP)
  Hello interval: Link 5 sec, Targeted 15 sec
  Hold time value sent in Hellos: Link 15 sec, Targeted 45 sec
  Keepalive interval: 6 sec, Hold time multiple: 6 intervals
```

Syntax: show mpls ldp

Table 219 lists the information displayed by the **show mpls ldp** command.

TABLE 219 Output from the show mpls ldp command

This field...	Displays...
Label Distribution Protocol version	The LDP version.
LSR ID	The identifier of the device and the loopback interface number being used by LDP. LDP advertises the address of this loopback interface in Address messages.
Hello interval	How often the device sends out LDP Link Hello and Targeted Hello messages.
Hold time value sent in Hellos	How long the device waits for its LDP peers to send a Hello message. The hold time is included in the Link Hello and Targeted Hello messages it sends out to its LDP peers.
Keepalive interval	The number of seconds between successive Keepalive messages send for an LDP session.
Hold time multiple	The number of Keepalive messages not received before a session is declared down.

Displaying information about LDP-created LSPs

You can display information about active LDP-created LSPs for which this device is an ingress, transit, or egress LSR.

Example

```
NetIron(config)# show mpls ldp path
Upstr-session(label)     Downstr-session(label, intf)    Destination route
33.3.3.3:0(3)            (egress)                        11.1.1.1/32
22.2.2.2:0(3)            (egress)                        11.1.1.1/32
33.3.3.3:0(1024)         22.2.2.2:0(3, e2/10)            22.2.2.2/32
22.2.2.2:0(1024)         22.2.2.2:0(3, e2/10)            22.2.2.2/32
(ingress)                22.2.2.2:0(3, e2/10)            22.2.2.2/32
33.3.3.3:0(1026)         33.3.3.3:0(3, e2/20)            33.3.3.3/32
22.2.2.2:0(1026)         33.3.3.3:0(3, e2/20)            33.3.3.3/32
(ingress)                33.3.3.3:0(3, e2/20)            33.3.3.3/32
```

Syntax: show mpls ldp path

Each line in the output of the **show mpls ldp path** command shows information about an LSP created through LDP. The command output lists the incoming and outgoing labels applied to packets in each LSP. For example, the third line in the example output indicates that MPLS packets received from upstream peer 33.3.3.3 with label 1024 are to be transmitted to downstream peer 22.2.2.2 with label 3.

NOTE
In this context, “upstream” and “downstream” shows the direction that data traffic flows in an LSP. This is opposite of the direction that labels are distributed using LDP.

Additionally, the output of this command indicates that the device has received a label for the destination IP prefix (that is, the attached route) from the downstream peer and then advertised a label for that IP prefix to the upstream peer.

Table 220 lists the information displayed by the **show mpls ldp path** command.

TABLE 220 Output from the **show mpls ldp path** command

This field...	Displays...
Upstr-session(label)	The LDP identifier of the upstream peer, as well as the incoming label. Note that upstream session information does not apply to LSPs for which this is the ingress LER. Because the device uses a per-platform label space, the incoming interface for LDP-created LSP is not relevant.
Downstr-session(label, intf)	The LDP identifier of the downstream peer, as well as the outgoing label and interface. When applicable, the ingress interface (intf) field displays a VE interface specified by the <*vid*> variable. Note that downstream session information does not apply to LSPs for which this is the egress LER. If LDP selects its outgoing interface as an RSVP tunnel, the ingress interface (intf) field displays the RSVP tunnel name.
Destination route	The destination route bound to this LSP.

Displaying LDP tunnel LSP information

To display information about LDP-created LSPs for which this device is the ingress LER, enter the following command.

```
NetIron# show mpls ldp tunnel
                Oper      Tunnel    Outbound
To              State     Intf      Intf
22.2.2.2        UP        tnl0      e3/1
33.3.3.3        UP        tnl1      e3/2
```

Syntax: show mpls ldp tunnel

The following table describes the output of the **show mpls ldp tunnel** command.

TABLE 221 Output from the **show mpls ldp tunnel** command

This field...	Displays...
To	The egress LER for the LSP.
Oper State	The operational state of the LSP. This field indicates whether the LSP has been established through LDP signalling and is capable of having packets forwarded through it.
Tunnel Intf	The MPLS tunnel interface port ID.
Outbound Intf	The outbound interface for the LSP. The outbound interface displays the egress interface of the tunnel. When applicable, the egress interface of the tunnel displays a VE interface specified by the <vid> variable.

Displaying the contents of the LDP database

You can display the contents of the LSR's LDP Label Information Base. This database contains all the labels it has learned from each of its LSR peers, as well as all of the labels it has sent to its LDP peers.

```
NetIron# show mpls ldp database
Session 1.1.1.1:0 - 2.2.2.2:0
 Downstream label database:
   Label     Prefix              State
   3         2.2.2.2/32          Installed
   1104      3.3.3.3/32          Retained
   1106      14.14.14.14/32      Retained
   1107      44.44.44.44/32      Retained
   800005    VC-FEC              Installed
 Upstream label database:
   Label     Prefix
   3         1.1.1.1/32
   1024      2.2.2.2/32
   1026      3.3.3.3/32
   1028      14.14.14.14/32
   1029      44.44.44.44/32
   800005    VC-FEC
   800006    VC-FEC
   983040    GEN ID VC-FEC
   983104    GEN ID VC-FEC
```

Syntax: show mpls ldp database

For each LDP session, the **show mpls ldp database** command displays the following information

TABLE 222 Output from the **show mpls ldp database** command

This field...	Displays...
Session	The LDP identifiers of this LSR and its peer.
Downstream label database	Information about labels received from the LDP peer
Upstream label database	Information about labels distributed by this LSR to the LDP peer. The device sends the same label for a given prefix to all of its upstream peers. In the example above, label 1028 is mapped to prefix 14.14.14.14/32. This device will send this label mapping to each of its upstream peers.
Label	The label value received from or distributed to LDP peers. It also displays the label values for VC FECs received from downstream LDP peers or advertised to up stream LDP peers.
Prefix	The destination route associated with the label. Since Prefix is not applicable to the VC-FECs, this field indicates that the label is associated with the VC FEC.
State	Whether the label is actively being used for data forwarding. This can be one of the following "Installed" indicates that the label is being used with an active LDP-created LSP to forward packets. "Retained" indicates that the label is not being used for packet forwarding. Since LSRs use Liberal Label Retention, these unused labels are retained in the database and not discarded.

Displaying LDP session information

To display information about the LDP session between this LSR and its LDP peers, enter the following command.

```
NetIron# show mpls ldp session
Peer LDP ID             State          Adj Used  My Role   Max Hold  Time Left
2.2.2.2:0               Operational    Link      Passive   36        32
3.3.3.3:0               Operational    Link      Passive   36        26
8.8.8.8:0               Operational    Targeted  Passive   36        33
14.14.14.14:0           Operational    Targeted  Passive   36        24
```

Syntax: show mpls ldp session

The following table describes the output of the **show mpls ldp session** command.

TABLE 223 Output from the **show mpls ldp session** command

This field...	Displays...
Peer LDP ID	The LDP identifier of the peer LSR. The first four octets identify the peer LSR IP address; the second two octets identify a label space on the LSR. For LSRs that use per-platform label spaces, the second two octets are always zero.
State	The current state of the LDP session between this LSR and its peer, as defined in RFC 3036. This can be "Nonexistent", "Initialized", "OpenRec", "OpenSent", or "Operational".
Adj Used	LDP adjacencies for a session: • Targeted • Link **NOTE:** When both link and targeted LDP adjacencies exist for a session, the adjacency type that is used will display Link. This is applicable on PowerConnect B-MLXe devices.
My Role	Whether this LSR is playing the “active” or “passive” role in LDP session establishment (as defined in RFC 3036). The LSR with the higher LSR ID plays the active role in LDP session establishment.
Max Hold	The number of seconds that the “Hold time remain” counter is reset to once a KeepAlive message is received from the peer.
Time Left	The amount of time, in seconds, before the LDP session times out if no KeepAlive message is received from the peer.

To display more detailed information about the LDP session between this LSR and its LDP peers, enter the **show mpls ldp session** command with the **detail** option as shown in the following.

```
NetIron# show mpls ldp session detail
Peer LDP ID: 120.120.120.2:0, Local LDP ID: 110.110.110.1:0, State: Operational
  Adj: Link, Role: Passive, Next keepalive: 4 sec, Hold time left: 37 sec
  Keepalive interval: 6 sec, Max hold time: 36 sec
  Up time: 6 min 51 sec
  MD5 Authentication Key: $MlVzZCFAbg==
  Neighboring interfaces: e2/11
  TCP connection: 110.110.110.1:646--120.120.120.2:9004, State: ESTABLISHED
  Next-hop addresses received from the peer:
    10.10.10.2  11.11.11.2  12.12.12.2  13.13.13.2  120.120.120.2
```

Syntax: show mpls ldp session detail

NOTE

The key displayed using the command, **show mpls ldp session detail** is the one currently configured for that session. This key may not be the one which is "in use" by that session as the session may have been established prior to the change in the configured key. If the session is already in the established state, any change in the authentication key will take effect during the next incarnation of the LDP session.

For each established LDP session, the command displays the following information:

TABLE 224 Output from the **show mpls ldp session detail** command

This field...	Displays...
Peer LDP ID	The LDP identifier of the peer LSR. The first four octets identify the peer LSR IP address; the second two octets identify a label space on the LSR. For LSRs that use per-platform label spaces, the second two octets are always zero.
Local LDP ID	This LSR's LDP identifier.
State	The LDP session state, as defined in RFC 3036. This can be "Nonexistent", "Initialized", "OpenRec", "OpenSent", or "Operational".
Adj	Whether the session was established using the Link adjacency (basic discovery) or the Targeted discovery (extended discovery).
Role	Whether this LSR is playing an "active" or "passive" role in session establishment.
Next keepalive	If this LDP session is established, the amount of time, in seconds, before the next KeepAlive message is sent to the active peer. If this LSR is the active peer, prior to establishing a session with the passive peer, the text "Next Initialization" is displayed instead. The "Next Initialization" value indicates, in seconds, when the next Initialization message will be sent to the passive peer.
Hold time left	The amount of time, in seconds, before the LDP session times out if no KeepAlive message is received from the peer.
Max hold time	The number of seconds that the "Hold time remain" counter is reset to once a KeepAlive message is received from the peer.
Up Time	The Up Time is displayed in days, hours, minutes, and seconds. This line appears only if the session is in Operational Up State.
Keepalive interval	The amount of time the LSR waits for an LDP PDU from the peer. If this amount of time passes without receiving an LDP PDU from the peer, the LDP session is terminated.
MD5 Authentication Key	The MD5 authentication key is displayed in an encrypted form when the user does not have the correct privileges. If the user has the correct permission, it will be displayed as clear text.
Neighboring interfaces	The interfaces where an LDP neighbor/adjacency relationship has been established with the peer. If there are multiple connections between two LDP-enabled peers, there can be multiple neighboring interfaces. When applicable, the Neighboring Interface field displays a VE interface specified by the <vid> variable.
TCP connection	The local and remote IP addresses and port numbers for the TCP connection between the peers.
State	The state of the TCP connection between the peers.
Next-hop addresses received from the peer	The next-hop addresses received from the peer in LDP address messages. The LSR uses this list of addresses to determine whether the peer is the correct next hop for a destination route. If one of the addresses in this list is the correct next hop for the route, the label received from the peer is installed for that route, allowing it to be used for data forwarding.

Displaying LDP neighbor connection information

To display information about the connection between this LSR and its LDP-enabled neighbors, enter the following command.

```
NetIron# show mpls ldp neighbor
Nbr Transport       Interface      Nbr LDP ID          Max Hold  Time Left
1.1.1.1             p4/1           1.1.1.1:0           15        14
5.5.5.5             p3/2           5.5.5.5:0           15        11
4.4.4.4             (targeted)     4.4.4.4:0           15        13
```

The **show mpls ldp neighbor** command is enhanced to include a detail option. The detail parameter includes Adjacency Up Time.

```
NetIron #show mpls ldp neighbor detail
Nbr Transport Addr: 22.22.22.1, Interface: e1/1, Nbr LDP ID: 22.22.22.1:0
 MaxHold: 44 sec, Time Left: 43 sec, Up Time: 36 min 22 sec

Nbr Transport Addr: 22.22.22.1, Interface: e1/2, Nbr LDP ID: 22.22.22.1:0
 MaxHold: 75 sec, Time Left: 74 sec, Up Time: 36 min 27 sec

Nbr Transport Addr: 33.33.33.1, Interface: e1/3, Nbr LDP ID: 33.33.33.1:0
 MaxHold: 75 sec, Time Left: 72 sec, Up Time: 36 min 22 sec

Nbr Transport Addr: 33.33.33.1, Interface: targeted, Nbr LDP ID: 33.33.33.1:0
 MaxHold: 75 sec, Time Left: 69 sec, Up Time: 35 min 36 sec
```

Syntax: show mpls ldp neighbor detail

TABLE 225 Output from the **show mpls ldp neighbor** command

This field...	Displays...
Nbr Transport	The transport address of the LDP neighbor.
Interface	The interface to which the LDP neighbor is connected. "(targeted)" indicates that the session between this device and the neighbor was established using Targeted Hello messages (that is, through extended discovery). When applicable, the Interface field displays a VE interface specified by the <vid> variable.
Nbr LDP ID	The neighbor's LDP identifier
Max Hold	The number of seconds the device waits for its LDP peers to send a Hello message.
Time Left	The amount of time, in seconds, before the LDP neighbor times out if no Hello message is received from the neighbor.
Up Time	The Up Time is the time since the LDP adjacency is established. It is displayed in days, hours, minutes, and seconds. If there is no Adjacency, then nothing is displayed

Displaying information about LDP-enabled interfaces

To display information about the LDP enabled interfaces on the LSR, enter the following command.

```
NetIron# show mpls ldp interface
                    Label-space    Nbr           Hello          Next
Interface           ID             Count         Interval       Hello
e4/1                0              1             5              --
(targeted)          0              0             15             --
```

Syntax: show mpls ldp interface

TABLE 226 Output from the **show mpls ldp interface** command

This field...	Displays...
Interface	The slot and port number of the LDP-connected interface. "(targeted)" shows information about unicast Targeted Hello messages sent to VLL peers.
Label-space ID	The label space ID. For LSRs that use per-platform label spaces, the second two octets are always zero.
Nbr Count	The number of LDP peers/adjacencies that has been established on this interface. This number can be greater than 1 if this is a multi-access network.
Hello Interval	The number of seconds between LDP Hello messages.
Next Hello	The number of seconds before the next LDP Hello message is sent (multicast) to the LDP interface (non-targeted). For a targeted interface, the LDP Hello message is unicast and, hence, for every neighbor, the next LDP Hello message is sent at a different time. In order to find out when the next LDP Hello message is sent out of any targeted adjacency, use the command **show mpls ldp neighbor**.

Displaying information about specified LDP-enabled nterface

To display information about a specific LDP enabled interface on the LSR, enter the following command.

```
NetIron# show mpls ldp interface ethernet 4/1
e4/1(Trunk1), label-space ID: 0
  Nbr count: 1
  Hello interval: 7 sec, next hello: 2 sec
  Hello timeout: 21 sec, hello timeout self: 0 sec
```

Syntax: show mpls ldp interface [ethernet <*slot/port*> **| pos** <*slot/port*>| **ve** <*vid*>**]**

TABLE 227 Output from the **show mpls ldp interface** command for a specific interface

This field...	Displays...
Interface	The slot and port number of the LDP-connected interface. The interface type refers to any one of the following: • **ethernet** <*slot/port*> to limit the display to a single ethernet port • **pos** <*slot/port*> to limt the display to a single pos port. • **ve** <*vid*> to limit the display to a VE interface ID specified by the <*vid*> variable.
Label-space ID	The label space ID. For LSRs that use per-platform label spaces, the second two octets are always zero.
Nbr Count	The number of LDP peers/adjacencies that has been established on this interface. This number can be greater than 1 if this is a multi-access network.
Hello Interval	The number of seconds between LDP Hello messages.

TABLE 227 Output from the **show mpls ldp interface** command for a specific interface

This field...	Displays...
Next Hello	The number of seconds before the next LDP Hello message is sent (multicast) to the LDP interface (non-targeted). For a targeted interface, the LDP Hello message is unicast and, hence, for every neighbor, the next LDP Hello message is sent at a different time. In order to find out when the next LDP Hello message is sent out of any targeted adjacency, use the command **show mpls ldp neighbor**.
Hello Timeout	The number of seconds that the interface waits for its LDP peers to send a Hello message. If the interface does not receive a Hello message within this time, the LDP session with the peer can be terminated.
Hello Timeout Self	The Hello Timeout Self configured for this interface. If the value for this parameter is set to 0, the Hello Timeout value is determined using the adjacent peer's sent hold timeout value. If that value is also set to 0, the global default value is used.

Displaying the LDP peer information

You can display LDP peering information as shown in the following.

```
NetIron# show mpls ldp peer
Peer LDP ID        State          Num-VLL        Num-VPLS-Peer
2.2.2.2:0          Operational    2              0
3.3.3.3:0          Operational    0              0
8.8.8.8:0          Operational    2              0
9.9.9.9:0          Unknown        2              0
14.14.14.14:0      Operational    1              0
```

Syntax: show mpls ldp peer <*peer-ip-address*> **| detail | brief]**

For each LDP session, the **show mpls ldp peer** command displays the following information

TABLE 228 Output from the **show mpls ldp peer** command

This field...	Displays...
Peer-addr	The LDP identifier of the peer LSR. The first four octets identify the peer LSR IP address; the second two octets identify a label space on the LSR. For LSRs that use per-platform label spaces, the second two octets are always zero.
State	The current state of the LDP session between this LSR and its peer.
Num-VLL	Number of VLL instances using this LDP peer.
Num-VPLS-Peer	Number of VPLS instances using this LDP peer.

To display more detailed information about the LDP peers, enter the following command.

```
NetIron#show mpls ldp peer detail
Peer LDP ID: 2.2.2.2:0, Local LDP ID: 1.1.1.1:0, State: Operational
Session Status UP, Entity Idx: 4, Targeted: No, Target Adj Added: Yes
Num VLL: 2, Num VPLS: 0
Rcvd VC-FECs:
 From 2.2.2.2: Label: 800001, VC Id: 120, Grp_Id: 0, VC Type: 4 MTU: 5000
```

```
Peer LDP ID: 8.8.8.8:0, Local LDP ID: 1.1.1.1:0, State: Operational
Session Status UP, Entity Idx: 2, Targeted: Yes, Target Adj Added: Yes
Num VLL: 2, Num VPLS: 0
Rcvd VC-FECs:
 From 8.8.8.8: Label: 16, VC Id: 19, Grp_Id: 0, VC Type: 32773, MTU: 5000
 From 8.8.8.8: Label: 18, VC Id: 18, Grp_Id: 0, VC Type: 32772, MTU 5555
```

For each LDP peer, the command displays the following information

TABLE 229 Output from the **show mpls ldp peer detail** command

This field...	Displays...
Peer LDP ID	The LDP identifier of the peer LSR. The first four octets identify the peer LSR IP address; the second two octets identify a label space on the LSR. For LSRs that use per-platform label spaces, the second two octets are always zero.
Local LDP ID	This LSR's LDP identifier.
State	The LDP session state, as defined in RFC 3036. This can be "Nonexistent", "Initialized", "OpenRec", "OpenSent", or "Operational".
Session Status	Whether the session is operationally UP or Down.
Entity Idx	This displays the LDP session entity CB index maintained by the LDP session controller.
Targeted	Whether the session was established using Targeted Hello messages (that is, through extended discovery).
Target Adj Added	Whether the targeted adjacency was initiated for this LDP peer.
Num VLL	Number of VLL instances using the LDP peer.
Num VPLS	Number of VPLS instances using the LDP peer.
Rcvd VC FECs	Displays the contents of received VC FECs
From	Peer LSR ID where the VC FEC was received from.
Label	The MPLS label associated with the VC FEC.
VC ID	The VC ID associated with the VC FEC.
Grp_ID	The Group ID associated with the VC FEC.
VC Type	The VC Type associated with the VC FEC.
MTU	The MTU value received in a VC Label Matching message from a peer.

To display detailed information about a specific LDP peer, enter the following command.

```
NetIron#show mpls ldp peer  22.22.22.22
Peer LDP ID: 22.22.22.22:0, Local LDP ID: 24.24.24.24:0, State: Operational
Session Status UP, Entity Idx: 1, Targeted: No, Target Adj Added: No
Num VLL: 0, Num VPLS: 0
Rcvd VC-FECs:
 From 5.5.5.5: Label: 100000, VC Id: 10, Grp_Id: 0, VC Type: 32773, MTU: 5000
 From 5.5.5.5: Label: 190448, VC Id: 200, Grp_Id: 0, VC Type: 32772, MTU 5555
```

For each LDP peer, the command displays the following information

TABLE 230 Output from the **show mpls ldp peer detail** command

This field...	Displays...
Peer LDP ID	The LDP identifier of the peer LSR. The first four octets identify the peer LSR IP address; the second two octets identify a label space on the LSR. For LSRs that use per-platform label spaces, the second two octets are always zero.
Local LDP ID	This LSR's LDP identifier.
State	The LDP session state, as defined in RFC 3036. This can be "Nonexistent", "Initialized", "OpenRec", "OpenSent", or "Operational".
Session Status	Whether the session is operationally UP or Down.
Entity Idx	This displays the LDP session entity CB index maintained by the LDP session controller.
Targeted	Whether the session was established using Targeted Hello messages (that is, through extended discovery).
Target Adj Added	Whether the targeted adjacency was initiated for this LDP peer.
Num VLL	Number of VLL instances using the LDP peer.
Num VPLS	Number of VPLS instances using the LDP peer.

Display considerations for LDP FEC information

The **show mpls ldp fec** command has changed to allow you to display all Layer 3 FEC information on the CLI, or specify the FEC type you want to display. When displaying the **show mpls ldp fec** command, consider the following.

- The prefix option is introduced to the **show mpls ldp fec** command. The **show mpls ldp fec prefix** command displays the total number of Layer 3 FECs. The total number of Layer 3 FECs is displayed in the Total number of prefix FECs field. For more information on this command, refer to "Displaying LDP FEC information" on page 1438.
- All options that are available under the **show mpls ldp fec** command have moved to the **show mpls ldp fec prefix** command.
- The **summary** option is introduced to the **show mpls ldp fec** command. The **show mpls ldp fec summary** command displays summarized FEC information. For more information on this command, refer to **"Displaying LDP FEC summary information"** on page 1440.
- The **vc-fec** option is renamed to **vc** option. All options that are under the **show mpls ldp vc-fec** command have moved under the **show mpls ldp fec vc** command. For more information on this command, refer to "Displaying the LDP FEC VC information" on page 1441.

Displaying LDP FEC information

To display host addresses and the total number of Layer 3 prefix FECs from the LDP FEC database, enter the following command.

```
NetIron# show mpls ldp fec prefix
Total number of prefix FECs: 2
Destination         State       Out-intf         Next-hop         Ingress Egress
125.125.125.1/32    current     e2/2             90.90.90.20      Yes     No
128.128.128.0/24    current     --               --               No      Yes
```

Syntax: show mpls ldp fec [<*IPaddress*> **[longer]** | **[prefix]**<*IPaddress_with_NetMask*>] **[longer]**

Using <*IPaddress*> |<*IPaddress_with_NetMask*> provides a detailed view of the specified FEC. Including the **longer** keyword means that if an IP address has been specified, display only the routes that match that IP address.The **prefix** option displays all Layer 3 FEC information.

The **show mpls ldp fec prefix** command displays the following information.

TABLE 231 Output from the **show mpls ldp fec prefix** command

This field...	Displays...
Total number of prefix FECs	The total number of Layer 3 FECs.
Destination	The IP Prefix associated with the host address or the prefix FEC type.
State	State of the FEC which indicates the FEC is currently being advertised to any LDP session (state equal to “current”). If it has no session, it will either be called “cur_no_sess” (currently no session) for local FECs or will be marked “retained” for non-local FECs.
Out-intf	For an ingress FEC, this mentions the output interface to reach to the Next-hop. The Out-Intf field displays the egress interface associated with the FEC entry. When applicable, the Out-Intf field displays a VE interface specified by the <*vid*> variable.
Next-hop	For an ingress FEC, this mentions the nexthop IP address.
Ingress	Whether the FEC is an ingress FEC.
Egress	Whether the FEC is an egress FEC.

Displaying information for a specified LDP FEC type

The **show mpls ldp fec prefix** <*IPaddress_with_NetMask*> command has changed. To display L3 FEC information for a specific FEC type, enter the following command.

```
NetIron#show mpls ldp fec prefix 125.125.125.1/32
FEC_CB: 0x29391f8c, idx: 1, type: 2, pend_notif: None
 State: current, Ingr: Yes, Egr: No, UM Dist. done: No
 Prefix: 125.125.125.1/32, next_hop: 90.90.90.20, out_if: e2/2

 Downstream mappings:
 Local LDP ID        Peer LDP ID         Label     State     CB
 128.128.128.28:0    125.125.125.1:0     3         Installe 0x29391cb0(-1)
```

Table 232 lists the output displayed for **show mpls ldp fec prefix** command.

TABLE 232 Output from the **show mpls ldp fec prefix** command

This field...	Displays...
FEC_CB	Memory address of the FEC CB.
idx	A monotonically increasing number assigned to each FEC in the LDP internal FEC tree.
type	FEC type – Prefix FEC is type 2 and Host Address is assigned type 3.
pend_notif	Any notification pending on this FEC.
State	State of the FEC which indicates the FEC is currently being advertised to any LDP session (state equal to “current”). If it has no session, it will either be called “cur_no_sess” (currently no session) for local FECs or will be marked “retained” for non-local.
Ingr	Whether the FEC is an ingress FEC.

TABLE 232 Output from the **show mpls ldp fec prefix** command (Continued)

This field...	Displays...
Egr	Whether the FEC is an egress FEC.
UM Dist	Specifies if Upstream Mapping Distribution is complete.
Prefix	The IP Prefix associated with the host address or the prefix FEC type.
next_hop	For an ingress FEC, this mentions the next- hop IP address. If LDP selects its outgoing interface as an RSVP tunnel, the next_hop field displays the RSVP tunnel destination address.
out_if	For an ingress FEC, this mentions the output interface to reach to the Next-hop. When applicable, the Out-Intf field displays a VE interface specified by the <*vid*> variable.
Downstream Mappings	Contents of the downstream mapping CB created as a result of the label mapping received from the downstream LDP peer.
Local LDP ID	Local LDP ID of the LDP session to which this downstream mapping CB belongs.
Peer LDP ID	Remote LDP ID of the LDP session to which this downstream mapping CB belongs.
Label	MPLS label received from the downstream LSR.
State	State of label. Either installed or retained.
CB	Memory address of the downstream mapping CB.

Displaying LDP FEC summary information

To display FEC summary information, enter the **show mpls ldp fec summary** command as shown in the following example.

```
NetIron#show mpls ldp fec summary
LDP FEC summary:
  Total number of prefix FECs: 8
  Total number of VC-FEC type 128: 0
  Total number of VC-FEC type 129: 0

LDP error statistics:
  Total number of route update processing errors: 0
  Total number of VC FEC processing errors: 0
```

Syntax: show mpls ldp fec summary

Table 233 lists the output displayed for the **show mpls ldp fec summary** command.

TABLE 233 Output from the **show mpls ldp fec summary** command

This field...	Displays...
LDP FEC summary	Summarized information for LDP FEC.
Total number of prefix FECs	The total number of prefix FECs in the LDP FEC database.
Total number of VC-FEC type 128	The total number of VC FECs for type 128. The FEC type for VC FEC can be 128 or 129.
Total number of VC-FEC TYPE 129	The total number of VC FECs for type 129. The FEC type for VC FEC can be 128 or 129.

This field...	Displays...
Total number of route update processing errors	The total number of route update processing errors for L3 FEC prefix.
Total number of VC FEC processing errors	The total number of L3 VC FEC internal processing errors.

Displaying the LDP FEC VC information

The **show mpls ldp vc-fec** command is renamed to **show mpls ldp fec vc** command. The output from the **show mpls ldp fec vc** command is enhanced to show the total number of VC FECs. The total number of VC FECs is displayed in the Total number of VC FECs field.

You can display a list of VC FECs from the LDP FEC database as shown in the following example.

```
NetIron#show mpls ldp fec vc
Total number of VC FECs: 2
Peer LDP ID           State      VC-ID      VC-Type FEC-Type Ingress Egress
125.125.125.1:0       current    100        4       128      Yes     Yes
125.125.125.1:0       current    1000       5       128      Yes     Yes
```

Syntax: show mpls ldp fec vc [<*vc-id*>]

The <*vc-id*> parameter provides a detailed view of the specified FEC VC. For more information on specifying the FEC VC, refer to "Displaying information for a specified LDP FEC VC" on page 1441.

The **show mpls ldp fec vc** command displays the following information.

TABLE 234 Output from the **show mpls ldp fec vc** command

This field...	Displays...
Total number of VC FECs	The total number of VC FECs.
Peer LDP ID	The remote LDP ID of the peer (or local LSR) where this VC FEC is originated from.
State	The state of the FEC which indicates the FEC is currently being advertised to any LDP session (state equal to "current"). If it has no session, it is either called "cur_no_sess" (currently no session) for local FECs or marked "retained" for non-local FECs.
VC-ID	The VC ID associated with the VC FEC.
VC-Type	The VC Type associated with the VC FEC.
FEC-Type	The number that identifies the FEC Type
Ingress	Whether the FEC in an ingress FEC.
Egress	Whether the FEC in an egress FEC.

Displaying information for a specified LDP FEC VC

The output from the **show mpls ldp fec vc** <*vc-id*> command has changed in the following:

- When a VLL or VPLS peer is up, only one FEC_CB is displayed in the output of the **show mpls ldp fec vc** <*vc-id*> command. Previously, two FEC_CBs were displayed in the output.

- The MTU enforcement field is introduced in the **show mpls ldp fec vc** <*vc-id*> command output. The MTU enforcement field indicates whether a MTU enforcement has been enabled. The MTU enforcement field, together with the Local MTU field and Remote MTU field indicates whether a MTU mismatch has occured.
- When the local and remote VC types for a specified VC ID do not match, two FEC_CBs are displayed.

The examples below describe these changes to the **show mpls ldp fec vc** <*vc-id*> command in more detail.

The following example displays VC FEC id 100 in an UP state. Since the local-mtu field and remote-mtu field both display the same MTU value of 1500 (MTU enforcement is enabled), the State field displays Installed. Local and Remote MTUs are now displayed together as part of one VC FEC_CB as shown in the example below.

```
NetIron#show mpls ldp fec vc 100
FEC_CB: 0x29391510, idx: 6, type: 128, pend_notif: None
 State: current, Ingr: Yes, Egr: Yes, UM Dist. done: Yes
 VC-Id: 100, vc-type: 4, grp-id: 0
 Local-mtu: 1500, remote-mtu: 1500, MTU enforcement: enabled

 Downstream mappings:
 Local LDP ID        Peer LDP ID         Label     State     CB
 128.128.128.28:0    125.125.125.1:0     800000    Installed 0x29391328(-1)

 Upstream mappings:
 Local LDP ID        Peer LDP ID         Label     CB
 128.128.128.28:0    125.125.125.1:0     800003    0x2939141c(-1)
```

The **show mpls ldp vc-fec** [<*vc-id*>] command displays the following information

TABLE 235 Output from the **show mpls ldp fec** command

This field...	Displays...
FEC_CB	Memory address of the FEC CB.
idx	A monotonically increasing number assigned to each FEC in the LDP internal FEC tree.
type	FEC type For VC FEC this value can be 128 or 129.
pend_notif	Any notification pending on this FEC.
State	State of the FEC which indicates the FEC is currently being advertised to any LDP session (state equal to “current”). If it has no session, it will either be called “cur_no_sess” (currently no session) for local FECs or will be marked “retained” for non-local FECs.
Ingr	Whether the FEC is an ingress FEC.
Egr	Whether the FEC is an egress FEC.
UM Dlst. done	Specifies if Upstream Mapping Distribution is complete.
VC-Id	The VC ID associated with the VC FEC.
vc-type	The VC Type associated with the VC FEC.
grp-id	The Group ID associated with the VC FEC.
Local-mtu	The local MTU for a specified VC FEC.
remote-mtu	The remote MTU for a specified VC FEC.
MTU enforcement	The user configured MTU enforcement setting that displays Enabled when a specified VC ID is up.

TABLE 235 Output from the **show mpls ldp fec** command (Continued)

This field...	Displays...
Downstream Mappings	Contents of the downstream mapping CB created as a result of the label mapping received from the downstream LDP peer.
Local LDP ID	Local LDP ID of the LDP session to which this downstream mapping CB belongs.
Peer LDP ID	Remote LDP ID of the LDP session to which this downstream mapping CB belongs.
Label	MPLS label received from the downstream LSR.
State	State of label. Either installed or retained.
CB	Memory address of the downstream mapping CB.
Upstream Mappings	Contents of the upstream mapping CB created as a result of the label mapping sent to the upstream LDP peer.
Local LDP ID	Local LDP ID of the LDP session to which this upstream mapping CB belongs.
Peer LDP ID	Remote LDP ID of the LDP session to which this upstream mapping CB belongs.
Label	MPLS label advertised to the upstream LDP LSR.
CB	Memory address of the upstream mapping CB.

The following example displays a MTU mismatch for VC ID of 100 where the VC label received from the remote peer is in a Retained state instead of an Installed state.

```
NetIron#show mpls ldp fec vc 100
FEC_CB: 0x293916f8, idx: 3, type: 128, pend_notif: None
 State: current, Ingr: Yes, Egr: Yes, UM Dist. done: Yes
 VC-Id: 100, vc-type: 4, grp-id: 0
 Local-mtu: 2000, remote-mtu: 1500, MTU enforcement: enabled

 Downstream mappings:
 Local LDP ID        Peer LDP ID         Label     State     CB
 128.128.128.28:0    125.125.125.1:0     800000    Retained  0x29391328(-1)

 Upstream mappings:
 Local LDP ID        Peer LDP ID         Label     CB
 128.128.128.28:0    125.125.125.1:0     800001    0x29391604(-1)
```

The following example displays a VC type mismatch where two VC FEC_CBs are displayed for the same VC ID of 1000. In the example, you can see that one VC type displays 5, and the other VC type displays 11. The two VC FEC_CBs are not associated with each other in any way. The VC type mismatch causes the VC label to display a Retained state instead of an Installed state.

```
NetIron#show mpls ldp fec vc 1000
FEC_CB: 0x29391234, idx: 5, type: 128, pend_notif: None
 State: retained, Ingr: Yes, Egr: Yes, UM Dist. done: No
 VC-Id: 1000, vc-type: 5, grp-id: 0, local-mtu: N/A, remote-mtu: 1500

 Downstream mappings:
 Local LDP ID        Peer LDP ID         Label     State     CB
 128.128.128.28:0    125.125.125.1:0     983040    Retained  0x29391140(-1)

FEC_CB: 0x29391510, idx: 4, type: 128, pend_notif: None
 State: current, Ingr: Yes, Egr: Yes, UM Dist. done: Yes
 VC-Id: 1000, vc-type: 11, grp-id: 0
 Local-mtu: 1500, remote-mtu: N/A, MTU enforcement: enabled

 Upstream mappings:
 Local LDP ID        Peer LDP ID         Label     CB
 128.128.128.28:0    125.125.125.1:0     983041    0x2939141c(-1)
```

Displaying the LDP packet statistics

You can display a packet statistics for packet types and packet errors, as shown in the following.

```
NetIron# show mpls ldp statistics
                      Total                Since last clear
 PacketType         Sent      Received       Sent      Received
  Link Hello         215       214           215        214
  Targeted Hello     138       110           138       110
  Init               1         1             1         1
  KeepAlive          16        18            16        18
  Notification       0         0             0         0
  Address            2         0             2         0
  AddressWithdraw    0         0             0         0
  LabelMapping       0         0             0         0
  LabelRequest       0         0             0         0
  LabelWithdraw      0         0             0         0
  LabelRelease       0         0             0         0
  LabelAbortReq      0         0             0         0

 Errors                       Total          Since last clear
  Rcv pkt bad pdu length      0              0
  Rcv pkt bad msg length      0              0
  Rcv pkt bad tlv length      0              0
  Rcv pkt notify unkn tlv     0              0
  Rcv pkt notify unkn addrfam 0              0
  Rcv pkt missing tlv         0              0
  Rcv pkt incorrect tlv       0              0
  Rcv pkt malformed tlv       0              0
  Rcv pkt bad traffic parm    0              0
  Rcv pkt partial pdu         0              0
  Rcv pkt internal error      0              0

  TCP send error              0              0
  TCP get send pkt error      0              0
  TCP memory fail             0              0
Num of TCP socket buffers: 0
```

Syntax: show mpls ldp statistics

The **show mpls ldp statistics** command displays the following information

TABLE 236 Output from the show mpls ldp statistics command

This field...	Displays...
PacketType	The type of LDP packet being counted.
Total	The number of the packets of the Type described for the row, sent and received since the router came up.
Since Last Clear	The number of the packets of the Type described for the row, sent and received since the last time a clear command was issued.
Errors	The type of packet error being counted. These errors are associated with the received packets only.
Total	The number of the errors of the Type described for the row, generated since the router came up.
Since Last Clear	The number of the errors of the Type described for the row, generated since the last time a clear command was issued.

Clearing the LDP packet statistics

You can clear the LDP Packet Statistics, as shown in the following commands.

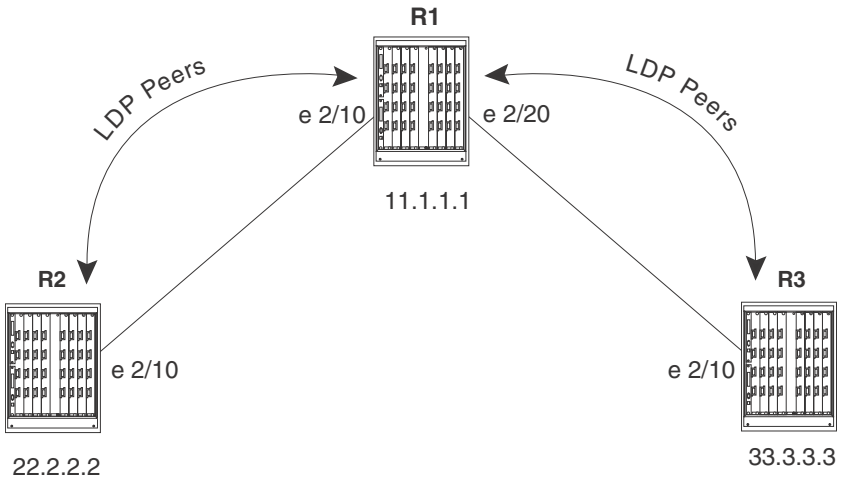
```
NetIron# clear mpls ldp statistics
```

Syntax: clear mpls ldp statistics

Sample LDP configurations

Figure 188 illustrates a sample configuration with three LDP-enabled LSRs.

FIGURE 188 Sample LDP configuration



Router R1

The following commands configure Router R1 in Figure 188.

```
R1(config)# interface loopback 1
R1(config-lbif-1)# ip address 11.1.1.1/32
R1(config-lbif-1)# exit
R1(config)# router mpls
R1(config-mpls)# mpls-interface e 2/10
R1(config-mpls)# ldp-enable
R1(config-mpls)# mpls-interface e 2/20
R1(config-mpls)# ldp-enable
R1(config-mpls)# exit
R1(config)# ip route 22.2.2.2/32 10.1.1.2
R1(config)# ip route 33.3.3.3/32 20.1.1.2
R1(config)# route-only
R1(config)# interface ethernet 2/10
R1(config-if-2/10)# enable
R1(config-if-2/10)# ip address 10.1.1.1/24
R1(config-if-2/10)# exit
R1(config)# interface ethernet 2/20
R1(config-if-2/20)# enable
R1(config-if-2/20)# ip address 20.1.1.1/24
```

Router R2

The following commands configure Router R2 in Figure 188.

```
R2(config)# interface loopback 1
R2(config-lbif-1)# ip address 22.2.2.2/32
R2(config-lbif-1)# exit
R2(config)# router mpls
R2(config-mpls)# mpls-interface e 2/10
R2(config-mpls)# ldp-enable
R2(config-mpls)# exit
R2(config)# ip route 11.1.1.1/32 10.1.1.1
R2(config)# ip route 33.3.3.3/32 10.1.1.1
R2(config)# route-only
R2(config)# interface ethernet 2/20
R2(config-if-2/20)# enable
R2(config-if-2/20)# ip address 10.1.1.2/24
R2(config-if-2/20)# exit
```

Router R3

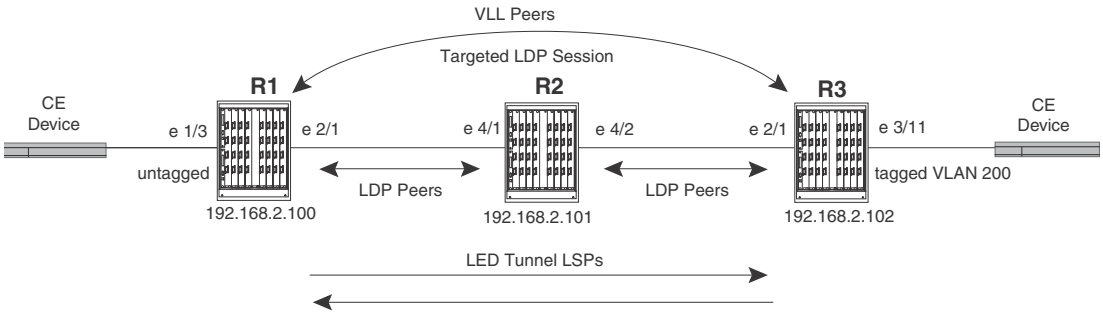
The following commands configure Router R3 in Figure 188.

```
R3(config)# interface loopback 1
R3(config-lbif-1)# ip address 33.3.3.3/32
R3(config-lbif-1)# exit
R3(config)# router mpls
R3(config-mpls)# mpls-interface e 2/10
R3(config-mpls)# ldp-enable
R3(config-mpls)# exit
R3(config)# ip route 11.1.1.1/32 20.1.1.1
R3(config)# ip route 22.2.2.2/32 20.1.1.1
R3(config)# route-only
R3(config)# interface ethernet 2/20
R3(config-if-2/20)# enable
R3(config-if-2/20)# 20.1.1.2/24
R3(config-if-2/20)# exit
```

Sample LDP configuration with VLL

Figure 189 illustrates a sample Virtual Leased Line (VLL) configuration that uses LDP tunnel LSPs.

FIGURE 189 MPLS VLL configuration with LDP tunnel LSPs



In this example, routers R1 and R3 are Provider Edge (PE) routers configured as VLL peers. R1 and R3 have established a targeted LDP session to exchange VLL label information. When this targeted LDP session is established, each router advertises its locally assigned VC label and VC ID to its VLL peer.

In addition, LDP sessions have been established between R1 – R2 and R2 – R3. LDP tunnel LSPs exist in each direction between R1 and R3. When the CE device forwards a Layer 2 packet to R1, the router assigns the packet to an LSP whose destination is R3. R1 encapsulates the packet as an MPLS packet, adding a tunnel label and the VC label advertised to the router by R3. The MPLS packet is then forwarded over the outbound interface indicated by the tunnel label to the next hop in the LSP.

When the MPLS packet reaches R2, the penultimate LSR in the tunnel LSP, R2 pops the tunnel label, leaving the packet with only the VC label, then forwards the packet to R3.

R3 examines the VC label in the packet. On R3, the VC label is mapped to the user-specified endpoint for the VLL. In this example, the endpoint consists of VLAN ID 200 and interface 3/11. R3 then pops the VC label, tags the Layer 2 packet with VLAN 200, then forwards the packet out interface 3/11.

In the opposite direction, R3 assigns traffic received from the CE device to a tunnel LSP destined for R1, pushes tunnel and VC labels onto the packets, and forwards them to the next hop in the LSP. When the packets reach R1, the router pops the VC label and forwards the Layer 2 packets out the interface indicated by the VLL endpoint. In this example, the endpoint consists of interface 1/3, so the packets are forwarded untagged out interface 1/3 to the CE device.

Router R1

The following commands configure Router R1 in Figure 189.

```
R1(config-mpls)# interface loopback 1
R1(config-lbif-1)# port-name Generic All-Purpose Loopback
R1(config-lbif-1)# ip address 192.168.2.100/32
R1(config-lbif-1)# ip ospf area 0
R1(config-lbif-1)# exit
R1(config)# router mpls
R1(config-mpls)# mpls-interface e 2/1
R1(config-mpls)# ldp-enable
R1(config-mpls)# exit
```

```
R1(config-mpls)# vll VLL_to_R3 40000
R1(config-mpls-vll)# vll-peer 192.168.2.102
R1(config-mpls-vll)# untagged e 1/3
R1(config-mpls-vll)# exit
R1(config)# ip router-id 192.168.2.100
R1(config)# router ospf
R1(config-ospf-router)# area 0
R1(config-ospf-router)# exit
R1(config-mpls)# interface e 1/3
R1(config-if-e100-1/3)# port-name VLL_endpoint
R1(config-if-e100-1/3)# enable
R1(config-if-e100-1/3)# exit
R1(config-mpls)# interface e 2/1
R1(config-e10000-2/1)# port-name Connection_to_R2
R1(config-e10000-2/1)# enable
R1(config-e10000-2/1)# ip address 192.168.37.1/30
R1(config-e10000-2/1)# ip ospf area 0
R1(config-e10000-2/1)# exit
```

Router R2

The following commands configure Router R2 in Figure 189.

```
R2(config-mpls)# interface loopback 1
R2(config-lbif-1)# port-name Generic All-Purpose Loopback
R2(config-lbif-1)# ip address 192.168.2.101/32
R2(config-lbif-1)# ip ospf area 0
R2(config-lbif-1)# exit
R2(config)# router mpls
R2(config-mpls)# mpls-interface e 4/1 e 4/2
R2(config-mpls)# ldp-enable
R2(config-mpls)# exit
R2(config)# ip router-id 192.168.2.101
R2(config)# router ospf
R2(config-ospf-router)# area 0
R2(config-ospf-router)# exit
R2(config-mpls)# interface e 4/1
R2(config-e10000-4/1)# enable
R2(config-e10000-4/1)# ip address 192.168.40.1/30
R2(config-e10000-4/1)# ip ospf area 0
R2(config-e10000-4/1)# exit
R2(config-mpls)# interface e 4/2
R2(config-e10000-4/2)# enable
R2(config-e10000-4/2)# ip address 192.168.40.9/30
R2(config-e10000-4/2)# ip ospf area 0
R2(config-e10000-4/2)# exit
```

Router R3

The following commands configure Router R3 in Figure 189.

```
R3(config-mpls)# interface loopback 1
R3(config-lbif-1)# port-name Generic All-Purpose Loopback
R3(config-lbif-1)# ip address 192.168.2.102/32
R3(config-lbif-1)# ip ospf area 0
R3(config-lbif-1)# exit
```

```
R3(config)# router mpls
R3(config-mpls)# mpls-interface e 2/1
R3(config-mpls)# ldp-enable
R3(config-mpls)# exit
R3(config-mpls)# vll VLL_to_R1 40000
R3(config-mpls-vll)# vll-peer 192.168.2.100
R3(config-mpls-vll)# vlan 200
R3(config-mpls-vll-vlan)# tagged e 3/11
R3(config-mpls-vll-vlan)# exit
R3(config-mpls-vll)# exit
R3(config)# ip router-id 192.168.2.102
R3(config)# router ospf
R3(config-ospf-router)# area 0
R3(config-ospf-router)# exit
R3(config-mpls)# interface e 3/11
R3(config-if-e100-3/11)# port-name VLL_endpoint
R3(config-if-e100-3/11)# enable
R3(config-if-e100-3/11)# exit
R3(config-mpls)# interface e 2/1
R3(config-e10000-2/1)# port-name Connection_to_R2
R3(config-e10000-2/1)# enable
R3(config-e10000-2/1)# ip address 192.168.41.1/30
R3(config-e10000-2/1)# ip ospf area 0
R3(config-e10000-2/1)# exit
```

31 Sample LDP configuration with VLL

Configuring MPLS Virtual Leased Line 32

Overview

The following list displays the MPLS Virtual Leased Line (VLL) features support by PowerConnect B-MLXe:

- MPLS Virtual Leased Line (VLL)
- MPLS VLL Packet Encoding
- QoS for VLL Traffic
- Tagged or Raw Mode for a VLL
- Dual tag support for MPLS VLL
- VLL MTU Enforcement
- VLL MTU
- Display changes to the **show mpls vll** detail command
- Local VLL
- VLAN Translation
- VPLS and VLL support - Per VLL MTU
- Dynamic LAG support for VLL endpoints
- Dual-tags for VLL-local
- MPLS Signalling: RSVP-TE support
- Traps for VLLs
- MPLS Local VLL Traps
- Disabling Syslog Messages for MPLS VLL-Local and VLL

This chapter explains how to configure **MPLS Virtual Leased Line (VLL)** on a NetIron. Virtual Leased Line is also known as Pseudo Wire Emulation as defined by the IETF PWE3 Working Group. MPLS VLL is a method for providing point-to-point Ethernet or VLAN connectivity over an MPLS domain. This functionality is outlined in the IETF documents “draft-ietf-pwe3-control-protocol-14.txt” and “draft-ietf-pwe3-ethernet-encap-08.txt”.

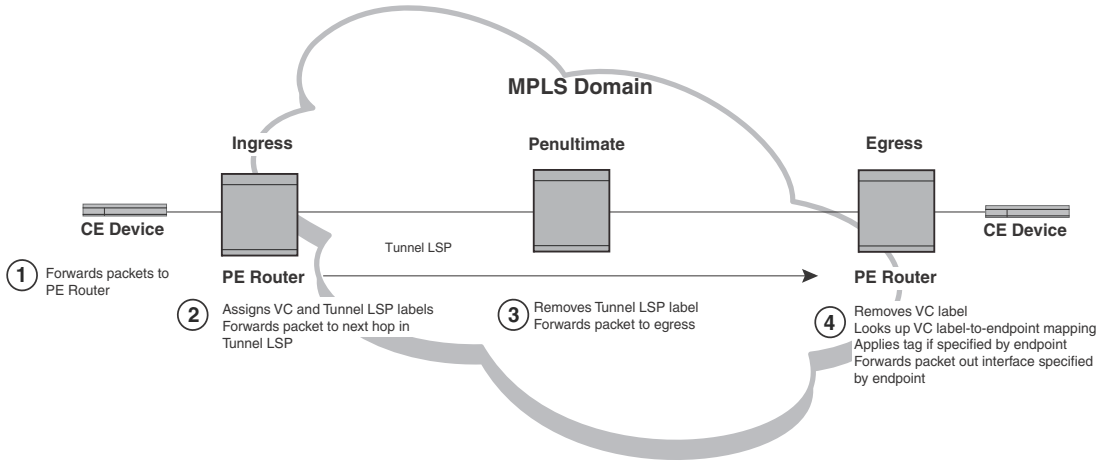
This chapter is divided into the following sections:

- “How MPLS VLL works” on page 1452 describes how packets are encapsulated and forwarded over an MPLS VLL.
- “Example 2: CoS behavior for dual-tagged to single-tagged VLL endpoints” on page 1459 describes how to set up MPLS VLLs on devices using the Command Line Interface (CLI).
- “Displaying MPLS VLL information” on page 1468 describes the commands used to display information about an MPLS VLL configuration.
- “Sample MPLS VLL configuration” on page 1477 illustrates a sample MPLS VLL configuration and lists the CLI commands used for implementing it.

How MPLS VLL works

The following diagram illustrates how packets are forwarded over an MPLS VLL.

FIGURE 190 Forwarding packets over an MPLS VLL



Packets are forwarded over an MPLS VLL as described below.

1. A Customer Edge (CE) device forwards a packet to a Label Edge Router (LER) serving as a Provider Edge (PE) router at the edge of the MPLS domain.
2. The PE router assigns the packet to an RSVP-signalled LSP whose destination is an LER (also serving as a PE router) that is connected to a CE device at the far end of the MPLS domain. The PE router at the other end of the MPLS domain is known as this PE router's ***VLL peer***. The RSVP-signalled LSP used to reach the VLL peer is known as the ***tunnel LSP***. Alternatively, an LDP-signalled, tunneled LSP can be used.

   If a Class of Service (COS) value is set for the VLL, the device selects a tunnel LSP that also has this COS value, if one is available. If no tunnel LSP with this COS value is available, the device selects a tunnel LSP with the highest configured COS value (although never higher than the COS setting for the VLL). Refer to "QoS for VLL traffic" on page 1453 for more information.

   If there are multiple tunnel LSPs that can be used to reach the VLL peer, the PE router selects one of the tunnel LSPs by using a round-robin method.

   The PE router pushes two labels onto the packet:

   - The inner ***VC label*** is used for determining what happens to the packet once it reaches the VLL peer. This label is significant only to the VLL peer.
   - The outer ***tunnel label*** is used for forwarding the packet through the MPLS domain. This label corresponds to an RSVP-signalled tunnel LSP.

   Refer to "MPLS VLL packet encoding" on page 1453 for information on the structure of packets forwarded along an MPLS VLL. After applying the two labels to the packet, the PE router forwards it to the next LSR in the tunnel LSP.
3. The penultimate LSR in the tunnel LSP removes the tunnel label and forwards the packet (now with the VC label as the top label) to the PE router at the other edge of the MPLS domain.
4. The VLL peer at the egress of the tunnel LSP examines the VC label. This VC label is mapped to an ***endpoint*** for the VLL. The endpoint of a VLL specifies what happens to packets exiting the VLL.

The endpoint can specify an untagged, dual-tagged, or single-tagged port.

- For *untagged* ports, the endpoint consists of an interface.
- For *single-tagged* ports, the endpoint consists of an interface and a VLAN ID.
- For *dual-tagged* ports, the endpoint consists of an interface and dual (outer and inner) VLAN IDs.

The egress LER removes the VC label and forwards the packet out the interface specified as the endpoint. If the endpoint is a single-tagged or dual-tagged port, the device transmits the packet with the specified VLAN ID, or with the dual tags, forwarding it out the specified interface to the CE device.

The two VLL peers advertise VC labels to each other using the ***Label Distribution Protocol (LDP)***. Each PE router attempts to initiate an LDP session with its VLL peer. After the LDP session is established, the locally assigned VC label, along with a VLL VC ID, is advertised to the VLL peer. In a similar way, the PE also learns the remotely assigned VC label from the VLL peer. Alternatively, you can configure static local and remote VC labels manually on both VLL peers; in this case, LDP is not used.

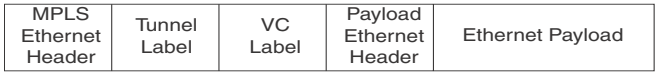
MPLS VLLs are not involved with spanning tree operations.

NOTE
If MTUs are mismatched on both sides of a VLL session, the session does not come up.

MPLS VLL packet encoding

When a packet is forwarded from the CE device, the PE router encapsulates it as an MPLS packet, applying two labels. The resulting MPLS packet has the following structure.

FIGURE 191 Structure of a packet forwarded over an MPLS VLL



The S bit in the tunnel label is zero, indicating that it is not the bottom of the stack. The VC label is significant only to the PE router at the other end of the VLL.

The Payload Ethernet header may be single-tagged or untagged.

QoS for VLL traffic

By default, packets travelling through an MPLS domain are treated equally from a QoS standpoint, in a best effort manner. However, if a Layer 2 packet has an internal priority in its 802.1q tag, or the LSP or VLL to which the packet is assigned has a configured Class of Service (COS) value, QoS can be applied to the packet in the MPLS domain. The internal priority or COS value is mapped to a value in the EXP field of the packet's MPLS header. The value in the EXP field is then mapped to an internal forwarding priority, and the packet is sent to the hardware forwarding queue that corresponds to the internal forwarding priority.

QoS for VLL traffic at the ingress LER

The following methods can be used to provide QoS to packets entering a VLL:

- Use the COS value assigned to the tunnel LSP used to reach the VLL peer.

When a tunnel LSP has a user-configured COS value, all packets in all VLLs travelling through the tunnel LSP receive the same QoS.

- Use the COS value assigned to the VLL.

  If a COS value is set for the VLL, the device selects a tunnel LSP that also has this COS value, if one is available. If no tunnel LSP with this COS value is available, the device selects a tunnel LSP with the highest configured COS value (although never higher than the COS setting for the VLL).

  If the selected tunnel LSP does not have a COS value, the VLL's configured COS value is used to provide QoS. The VLL's COS value is mapped to a value in the EXP field. This allows traffic multiple VLLs using a single tunnel LSP, traffic from each VLL can receive different QoS treatment.

- Use the priority in the packet's 802.1q tag.

  When neither the tunnel LSP nor the VLL has a configured COS value, the device examines the priority in the Layer 2 packet's 802.1q tag, if the packet has one. Consequently, Layer 2 packets with the same 802.1q priority receive the same QoS in the VLL.

- Use the configured priority of the port.

  If neither the tunnel LSP nor the VLL has a configured COS value, and the Layer 2 packet does not have an 802.1q priority, QoS can be provided based on the priority of the incoming port. A port can be assigned a priority from 0 (lowest priority) to 7 (highest priority). The default port priority is 0.

  By assigning different priorities to the ports where customer edge (CE) devices are connected (that is, the VLL endpoints), you can provide QoS to untagged Layer 2 traffic received from different customer locations.

When a packet enters a VLL, the PE router that serves as both the VLL endpoint and the ingress of a tunnel LSP pushes two labels onto the packet the inner VC label and the outer tunnel label. The packet's priority resides in the EXP field of the MPLS label header. The VC label and the tunnel label carry the same value in the EXP field.

The following table lists how a Layer 2 packet's priority is mapped to a value in the EXP field and how the EXP value is mapped to a priority queue.

Tunnel LSP configured COS or VLL configured COS or 802.1q priority or Configured port priority	Value placed in the tunnel and VC label EXP field	Priority queue
7	7	qosp7 (highest priority)
6	6	qosp6
5	5	qosp5
4	4	qosp4
3	3	qosp3
2	2	qosp2
1	1	qosp1
0	0	qosp0 (best effort)

QoS for VLL traffic at transit LSRs

At each transit LSR, the device reads the value in the tunnel label's EXP field and places the incoming EXP value in the EXP field of the outbound packet. The outbound MPLS packet is assigned to one of the eight priority queues based on the value in the EXP field. The EXP bits in the MPLS header are used to assign the packet to a priority queue as follows:

EXP Bits in tunnel label	Priority queue
7	qosp7 (highest priority)
6	qosp6
5	qosp5
4	qosp4
3	qosp3
2	qosp2
1	qosp1
0	qosp0 (best effort)

QoS for VLL traffic at the penultimate LSR

When the packet reaches the penultimate LSR in the LSP, its tunnel label is popped, leaving the VC label. The MPLS packet is placed in one of the priority queues using the value in the EXP field of the VC label. Since the VC label has the same EXP value as the tunnel label, the packet is placed in the same queue used for the tunnel LSP.

QoS for VLL traffic at the egress LER

At the VLL endpoint, the VC label is popped and the packet is forwarded as a Layer 2 packet. The packet is placed in one of the priority queues based on the contents of the EXP field in the VC label, as follows:

EXP Bits in VC label	Priority queue
6, 7	qosp3 (highest priority)
4, 5	qosp2
2, 3	qosp1
0, 1	qosp0 (best effort)

CoS behavior for VLL tagged mode and VLL raw mode

This section describes the difference in CoS behavior for VLL traffic when tagged mode or raw mode is in effect.

CoS behavior for VLL tagged mode

NOTE
This section assumes that you understand how QoS works. For details, see 9, "Configuring Quality of Service for the NetIron MLX".

Table 237 describes the expected Class of Service (CoS) behavior for VLL packets when VLL tagged mode is enabled.

TABLE 237 Expected class of service behavior for VLL tagged mode

VLL endpoints	Incoming packet		MPLS cloud		Outgoing packet	
	Outer VLAN	Inner VLAN	Tunnel/VC label (Z)	Payload tag	Outer VLAN	Inner VLAN
Dual-tagged to dual-tagged	X	Y	V or internal priority	Y	W or Y	Y
Single-tagged to dual-tagged	X	N/A		X	W or X	X
Untagged to dual-tagged	N/A	N/A		0	W or 0	0
Dual-tagged to single-tagged	X	Y		Y	W or Y	N/A

NOTE
For more specific examples of CoS behavior for tagged mode, see "Example 1: CoS behavior for dual-tagged to dual-tagged VLL endpoints" on page 1457 and "Example 2: CoS behavior for dual-tagged to single-tagged VLL endpoints" on page 1459.

Legend for Table 237

V = mapped EXP bits from internal priority (X contributes to internal priority) using the EXP encode table. Table 57 shows the default EXP encode table.

W = mapped COS from internal priority (Z contributes to internal priority) using the COS encode table

X = original outer VLAN COS

Y = original inner VLAN COS

Z = incoming EXP bits as described by Tunnel / VC label column = V or internal priority

or in the *Tunnel/VC label* column differentiates the behavior between when **qos exp encode** policy is ON (default) or OFF.

or in the *Outgoing packet Outer VLAN* column differentiates the behavior between when **qos pcp encode** policy is ON (default) or OFF.

CoS behavior for VLL raw mode

NOTE
This section assumes that you understand how QoS works. For details, see 9, "Configuring Quality of Service for the NetIron MLX".

Table 238 describes the expected Class of Service (CoS) behavior for VLL packets when VLL raw mode is in effect.

TABLE 238 Expected class of service behavior for VLL raw mode

VLL endpoints	Incoming packet		MPLS cloud		Outgoing packet	
	Outer VLAN	Inner VLAN	Tunnel/VC label (Z)	Payload tag	Outer VLAN	Inner VLAN
Dual-tagged to dual-tagged	X	Y	V or internal priority	N/A	W or Z	Z
Single-tagged to dual-tagged	X	N/A				Z
Untagged to dual-tagged	N/A	N/A				Z
Dual-tagged to single-tagged	X	Y				N/A

NOTE
For more specific examples of CoS behavior for raw mode, see "Example 1: CoS behavior for dual-tagged to dual-tagged VLL endpoints" on page 1457 and "Example 2: CoS behavior for dual-tagged to single-tagged VLL endpoints" on page 1459.

Legend for Table 238

V = mapped EXP bits from internal priority (X contributes to internal priority) using the EXP encode table. Table 53 shows the default EXP encode table.

W = mapped COS from internal priority (Z contributes to internal priority) using the COS encode table.

X = original outer VLAN COS

Y = original inner VLAN COS

Z = incoming EXP bits as described by Tunnel / VC label column = V or internal priority

or in the *Tunnel/VC label* column differentiates the behavior when **qos exp encode** policy is ON (default) or OFF.

or in the *Outgoing packet Outer VLAN* column differentiates the behavior when **qos pcp encode** policy is ON (default) or OFF.

Example 1: CoS behavior for dual-tagged to dual-tagged VLL endpoints

Table 239 shows a detailed example of the CoS behavior in a dual-tagged to dual-tagged VLL endpoint configuration. The table shows the difference in behavior for VLL tagged mode (described in Table 237) versus VLL raw mode (described in Table 238).

TABLE 239 Example CoS behavior in a dual-tagged to dual-tagged VLL endpoint configuration

Priority	Incoming Packet		Outgoing Packet Tag Mode		Outgoing Packet Raw Mode	
	Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN
LSP CoS 4	VLAN 100, CoS 0	VLAN 200, CoS 0	VLAN 300, CoS 4	VLAN 400, CoS 0	VLAN 300 CoS 4	VLAN 400 CoS 4

TABLE 239 Example CoS behavior in a dual-tagged to dual-tagged VLL endpoint configuration

VLL CoS 2	VLAN 100, CoS 0	VLAN 200, CoS 0	VLAN 300, CoS 2	VLAN 400, CoS 0	VLAN 300 CoS 2	VLAN 400 CoS 2
Port priority 6 (with priority force)	VLAN 100, CoS 0	VLAN 200, CoS 0	VLAN 300, CoS 6	VLAN 400, CoS 0	VLAN 300 CoS 6	VLAN 400 CoS 6
802.1p CoS 6 (outer VLAN) CoS 4 (inner VLAN)	VLAN 100, CoS 6	VLAN 200, CoS 4	VLAN 300, CoS 6	VLAN 400, CoS 4	VLAN 300 CoS 6	VLAN 400 CoS 6
Port priority 6 and VLL CoS 2	VLAN 100, CoS 0	VLAN 200, CoS 0	VLAN 300, CoS 2	VLAN 400, CoS 0	VLAN 300 CoS 2	VLAN 400 CoS 2
Port priority 5 (with priority force)	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 5	VLAN 400, CoS 7	VLAN 300 CoS 5	VLAN 400 CoS 5
Port priority 5 (with priority force), LSP CoS 3	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 3	VLAN 400, CoS 7	VLAN 300 CoS 3	VLAN 400 CoS 3
Port priority 5 (with priority force), LSP CoS 2. VLL CoS 4	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 2	VLAN 400, CoS 7	VLAN 300 CoS 2	VLAN 400 CoS 2
Port priority 5 (with priority force), LSP CoS 0. VLL CoS 4	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 0	VLAN 400, CoS 7	VLAN 300 CoS 0	VLAN 400 CoS 0
Port priority 5 (with priority force), LSP no value. VLL CoS 4	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 4	VLAN 400, CoS 7	VLAN 300 CoS 4	VLAN 400 CoS 4
Port priority 5 (with priority force), LSP CoS 0. VLL CoS 4 QoS exp encode policy all-zero (ingress router)	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 0	VLAN 400, CoS 7	VLAN 300 CoS 0	VLAN 400 CoS 0
Port priority 5 (with priority force), LSP CoS 3. VLL CoS 4 QoS exp encode policy all-zero (ingress router)	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 0	VLAN 400, CoS 7	VLAN 300 CoS 0	VLAN 400 CoS 0
Port priority 5 (with priority force), LSP CoS 3. VLL CoS 4 QoS PCP encode policy all-zero (egress router)	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 0	VLAN 400, CoS 7	VLAN 300, CoS 0	VLAN 400, CoS 3
Port priority 5 (with priority force), LSP CoS 3. VLL CoS 4 QoS PCP decode policy string (ingress router) (mapping of 7 to 1)	VLAN 100, CoS 7	VLAN 200, CoS 6	VLAN 300, CoS 3	VLAN 400, CoS 6	VLAN 300, CoS 3	VLAN 400, CoS 3
QoS PCP decode policy string (ingress router) (mapping of 7 to 1)	VLAN 100, CoS 7	VLAN 200, CoS 6	VLAN 300, CoS 1	VLAN 400, CoS 6	VLAN 300, CoS 1	VLAN 400, CoS 1

Example 2: CoS behavior for dual-tagged to single-tagged VLL endpoints

Table 240 shows a detailed example of the CoS behavior in a dual-tagged to single-tagged VLL endpoint configuration. The table shows the difference in behavior for VLL tagged mode (described in Table 237) versus VLL raw mode (described in Table 238)..

TABLE 240 Example CoS behavior in a dual-tagged to single-tagged VLL endpoint configuration

Priority	Incoming Packet		Outgoing Packet Tag Mode		Outgoing Packet Raw Mode	
	Outer VLAN	Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN	Outer VLAN
LSP CoS 4,	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 4	NA	VLAN 300 CoS 4	NA
VLL CoS 2	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 2	NA	VLAN 300 CoS 2	NA
Port priority 7 (with priority force)	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 7	NA	VLAN 300 CoS 7	NA
802.1p CoS 6 (outer VLAN)	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 6	NA	VLAN 300 CoS 6	NA
Port priority 7 and VLL CoS 2	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 2	NA	VLAN 300 CoS 2	NA
Port priority 7 (with priority force), LSP CoS 3	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 3	NA	VLAN 300 CoS 3	NA
Port priority 7 (with priority force), LSP CoS 3. VLL CoS 4	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 3	NA	VLAN 300 CoS 3	NA
Port priority 7 (with priority force), LSP CoS 0. VLL CoS 4	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 0	NA	VLAN 300 CoS 0	NA
.1p is 6 for outer VLAN, 5 for inner VLAN Port 3 No LSP CoS VLL CoS 4 (ingress above) Egress below Port is 7 VLL CoS 2	VLAN 100 CoS 6	VLAN 200 CoS 5	VLAN 300 CoS 7	NA	VLAN 300 CoS 7	NA
Port priority 7 (with priority force), LSP no value. VLL CoS 4	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 4	NA	VLAN 300 CoS 4	NA
Port priority 7 (with priority force), LSP CoS 3. VLL CoS 4	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 3	NA	VLAN 300 CoS 3	NA
Port priority 7 (with priority force), LSP CoS 3. VLL CoS 4 QoS exp encode policy all-zero (ingress router)	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 0	NA	VLAN 300 CoS 0	NA

TABLE 240 Example CoS behavior in a dual-tagged to single-tagged VLL endpoint configuration

Port priority 7 (with priority force), LSP CoS 3. VLL CoS 4 QoS PCP encode policy all-zero (egress router)	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 0	NA	VLAN 300, CoS 0	NA
Port priority 6 (with priority force), LSP CoS 3. VLL CoS 4 QoS PCP decode policy string (ingress router) (mapping of 7 to 1)	VLAN 100, CoS 7	VLAN 200 CoS 5	VLAN 300, CoS 3	NA	VLAN 300, CoS 3	NA
QoS PCP decode policy string (ingress router) (mapping of 7 to 1)	VLAN 100, CoS 7	VLAN 200 CoS 5	VLAN 300, CoS 1	NA	VLAN 300, CoS 1	NA

Configuring MPLS VLLs

This section explains how to set up MPLS VLLs.

Creating a VLL

You create a VLL by entering VLL configuration statements on two PE routers. The two endpoints of a VLL are associated by having the same VLL VC ID on each PE router.

To create an MPLS VLL, enter commands such as the following.

```
NetIron(config-mpls)# vll foundry-sj-to-sf 40000
NetIron(config-mpls-vll)#
```

On the VLL peer (if it is a device), you would enter commands such as the following.

```
NetIron(config-mpls)# vll foundry-sf-to-sj 40000
NetIron(config-mpls-vll)#
```

Syntax: vll <*vll-name*> <*vll-vc-id*> **[cos** <*cos value*>**] [raw-mode]**

The <*vll-vc-id*> corresponds to the user-configurable ID defined in draft-ietf-pwe3-control-protocol-14.txt.

You can optionally specify a Class of Service (COS) setting for the VLL. If a COS value is set, the device selects a tunnel LSP that also has this COS value, if one is available. If no tunnel LSP with this COS value is available, the device selects a tunnel LSP with the highest configured COS value (although never higher than the COS setting for the VLL). The COS value can be between 0 – 7.

Specifying tagged or raw mode for a VLL

The default treatment for packets that are sent through a VLL is for the ingress router to add a VLAN ID tag to the payload Ethernet header. When the packet arrives at the egress router, the tag is stripped off and the packet is forwarded.

You can configure a VLL to send all packets in “raw” mode. This means that the ingress router of the VLL will not add a VLAN ID tag to the payload Ethernet header and consequently the egress router will not have to strip it off. Both the ingress and egress routers must be configured in either default (tagged mode) or raw mode. To configure a router to send or receive packets for a VLL in raw mode, enter the following command.

```
NetIron(config-mpls)# vll <vll-name> <vll-vc-id> raw-mode
```

Syntax: vll <*vll-name*> <*vll-vc-id*> **[raw-mode]**

The <*vll-name*> is the name of the VLL you want to configure raw mode for.

The <*vll-vc-id*> corresponds to the user-configurable ID defined in draft-ietf-pwe3-control-protocol-14.txt.

If **raw-mode** is specified, the VC type for signaling will be 5, otherwise it will be 4 (for tagged mode).

If **raw-mode** is not specified, the default configuration is for the ingress router to send packets with a tag, and for the egress router to strip it off before forwarding the packets.

NOTE
If there is a VC type mismatch between VLL peers, a session will not be brought up between them.

Specifying a VLL peer

The VLL peer is the PE router at the other end of the VLL. As part of VLL configuration, you specify the IP address of the VLL peer.

Each PE router must have tunnel LSP reachability to its VLL peer. Tunnel LSP reachability is defined as having at least one operational LSP tunnel with the destination (the LSP’s “to” address) matching the VLL peer’s IP address. An LSP terminating on the VLL peer but configured with a different destination address would not be considered a match.

If a PE router does not have tunnel LSP reachability to its VLL peer, or if the remote VC label is not yet available, packets from the local interface are discarded at the ingress PE router. If the local interface is administratively disabled or goes down, a VC label withdraw message is sent to the VLL peer.

By default, each PE router attempts to initiate an LDP session through extended discovery with its VLL peer, if a session is not already established. The PE router also allocates a VC label from a per-platform label range that is mapped to the local endpoint. Once the LDP session is established, the locally assigned VC label, along with the VLL VC ID is advertised to the VLL peer in a downstream-unsolicited manner. In a similar way, the PE also learns the remotely assigned VC label from the VLL peer.

Alternatively, you can configure static local and remote VC labels. In this case, no LDP session is established between the VLL peers. Note that if you use static VC labels, you must configure them on both VLL peers manually.

You specify the peer at the other end of the VLL by entering a command such as the following.

```
NetIron(config-mpls-vll)# vll-peer 192.168.2.100
```

Syntax: vll-peer <*ip-addr*> [<*static-local-vc-label*> <*static-remote-vc-label*>]

The IP address of the peer must match that of a destination for a tunnel LSP configured on the device.

Static local and remote VC label values are optional. If configured, <*static-local-vc-label*> is the VC label value expected for packets forwarded to the local physical port from the VLL peer, and <*static-remote-vc-label*> is the VC label applied to packets sent to the remote VLL peer.

Acceptable values for <*static-local-vc-label*> are 800000 – 1048575. If the label value you specify has already been assigned, a message is displayed requesting a different value.

Specifying a VLL endpoint

The endpoint of a VLL specifies what happens to packets exiting the VLL. You set the endpoint on the local PE router and this endpoint is mapped to a VC label. The VC label is advertised to the remote PE router at the other end of the VLL through LDP. The remote PE router applies this label to packets entering the VLL. When the packet reaches the end of the VLL through the MPLS uplink, the local PE router checks the mapping between the VC label and the endpoint, removes the VC label from the packet, and forwards the packet out the port specified as the endpoint.

All VLL endpoints can be dual-mode ports (tagged-untagged). An untagged endpoint port is removed from the default VLAN and cannot be added back to the default VLAN. A VLL endpoint can be tagged in multiple VLL and L2 VLANs and untagged in one other VLAN.

The Customer Edge (CE) device is connected to the PE router over an untagged, dual-tagged, or single-tagged port.

- With a *single-tagged* port, each pair (port, VLAN ID) is identified as a unique endpoint, and the packets are sent in tagged Ethernet format.
- In the case of an *untagged* port, an endpoint is identified by the physical port alone, and the packets are sent in untagged Ethernet format.
- In the case of a *dual-tagged* port, the packets contain both an outer VLAN tag and an inner VLAN tag.

Special considerations for VLL dual-tagged endpoints

Before configuring a dual-tagged endpoint, consider the following:

- An Internal Forwarding Lookup Identifier (IFL-ID) will be allocated to each MPLS VLL instance that has a dual-tagged endpoint. The ID will be displayed in the **show mpls vll detail** command output. For instances that do not have dual-tagged endpoints, the IFL-ID will be displayed as '--'.

  NOTE
  The acceptable values and configuration procedures for the system-max ifl-cam command are in the section "Configuring system max values" on page 106.

- The tag protocol identifier (TPID) of the inner VLAN tag must be 0x8100 in order to be classified as dual-tagged and recognized by dual-tagged endpoints. If the TPID is not 0x8100, the packet will be classified as a single-tagged packet.
- The same port, outer VLAN, and inner VLAN combination cannot be specified across MPLS VLL instances. For example, if a dual-tagged endpoint with vlan 100 and inner-vlan 200 is configured on port e 2/1 in MPLS VLL instance 'test', the same endpoint cannot be configured as part of another MPLS VLL instance, say 'test1'. This is also true across applications. That is, if a port, outer VLAN, and inner VLAN combination belongs to a MPLS VLL instance, it cannot simultaneously belong to a Layer 2 VLAN, Local VLL or VPLS.
- To change an existing single-tagged VLL endpoint to a dual-tagged endpoint, first delete the VLAN configuration, then configure the endpoint as dual-tagged.

- A dual-tagged VLL endpoint neither recognizes nor forwards packets that have a single tag. However, a single-tagged endpoint can recognize and forward dual-tagged packets because the endpoint treats the second tag as data.
- The port, outer VLAN, and inner VLAN combination in an incoming dual- tagged packet on a given port will be used to do an IFL CAM lookup. This lookup will yield an IFL-ID which will be used to do a MPLS-VLL CAM lookup. So for dual-tagged endpoints, the regular (port, vlan) lookup is replaced with the (port, IFL-ID) lookup.
- If only the outer VLAN is specified for a given endpoint, it is called a *less-specific* VLAN. If both the outer and inner VLAN are specified, it is called a *more-specific* VLAN (in relation to the outer VLAN).
- If a less-specific VLAN is already configured on a given port, then a more-specific VLAN with the same outer VLAN tag can also be configured on that port. Likewise, when a more-specific VLAN is already configured on a given port, then a less-specific VLAN with the same outer VLAN tag can also be configured on the port.

  In the following example, a less-specific tagged endpoint has been configured with vlan 100 on port e 2/1, and a more-specific VLAN with an outer VLAN tag of **100** and an inner vlan tag of **200** has also been configured on port e 2/1.

```
MLXe(config-mpls)#vll test1 1000
MLXe(config-mpls-vll-test1)#vlan 100
MLXe(config-mpls-vll-test1-vlan)#tag e 2/1
MLXe(config-mpls-vll-test1-if-e-2/1)#vll test2 2000
MLXe(config-mpls-vll-test2)#vlan 100 inner-vlan 200
MLXe(config-mpls-vll-test2-vlan)#tag e 2/1
```

  This applies even when the less/more-specific VLAN is configured as part of a L2 VLAN, Local VLL or VPLS.

Specifying an untagged endpoint

Untagged ports are not associated with any VLAN. A port must be a member of the default VLAN before it can be used in a VLL configuration as an untagged port. Upon configuration as the endpoint of a VLL, the port is taken out of the default VLAN. This means no local broadcast traffic includes this port. A VLL untagged port does not belong to any VLAN. If the port is currently a member of a regular VLAN or another VLL, the configuration attempt should be rejected.

NOTE
If a port is added as an untagged port into a VLL, a VLAN should not be defined under the VLL. If a VLAN is configured under the VLL, the configuration to add an untagged port will be rejected.

To specify an untagged endpoint for a VLL.

```
NetIron(config-mpls-vll)# untagged e 2/1
```

Syntax: untagged [ethernet] <*portnum*>

NOTE
Foundry Discovery Protocol (FDP) should not be enabled on an untagged VPLS or VLL endpoint.

Specifying a single-tagged endpoint

Tagged ports are configured under a VLAN ID. This VLAN ID is only meaningful for this tagged port. Another tagged port may use the same VLAN ID but the two ports are not under the same VLAN.

For tagged ports, a <*vlan-id, port*> pair constitutes a VLL endpoint. One VLL may configure a VLAN ID on one port and another VLL may reuse the same VLAN ID on another port. This capability is know as VLAN ID reuse.

As with regular VLANs, if a port is currently a member of a non-default VLAN as an untagged port, it must be returned to the default VLAN before it can be assigned to a VLL as a tagged port.

To specify a tagged endpoint for a VLL.

```
NetIron(config-mpls-vll)# vlan 200
NetIron(config-mpls-vll-vlan)# tagged e 3/11
```

Syntax: vlan <*num*>

Syntax: tagged [ethernet] <*slot/port*>

NOTE
A tagged port can be a part of one or more VLLs, and at the same time be part of one or more regular VLANs and one or more VPLSs as a tagged member.

Specifying a dual-tagged endpoint

Dual-tagged packets contain both an outer VLAN tag and an inner VLAN tag. Dual-tagged VLL endpoints enable MPLS VLL to recognize packets with two tags and make forwarding decisions based on them. A dual-tagged endpoint can receive packets with two tags and forward them to the other endpoint either untagged, single-tagged, or dual-tagged.

NOTE
Before configuring a dual-tagged endpoint, see "Special considerations for VLL dual-tagged endpoints" on page 1462.

To specify a dual-tagged endpoint for a VLL instance, enter commands such as the following:

```
NetIron(config-mpls)# vll test 100
NetIron(config-mpls-vll-test)# vlan 200 inner-vlan 300
NetIron(config-mpls-vll-test-vlan)#tagged e 3/11
```

Syntax: [**no**] **vlan** <*vlan-id*> **inner-vlan** <*vlan-id*>

Syntax: [**no**] **tagged ethernet** <*slot/port*>

The **vlan** <*vlan-id*>, which is the outer VLAN ID, can be in the range from 1 to 4094 and excludes the default VLAN.

The **inner-vlan** <*vlan-id*> can be in the range from 1 to 4095 and includes the default VLAN.

Use the **no** form of the command to remove the dual-tagged VLL VLAN configuration and its associated endpoints. For example, the command **no vlan 200 inner-vlan 300** will remove the dual-tagged VLAN and associated endpoints. The single-tagged VLAN, **vlan 200**, will not be deleted. Similarly, the command **no vlan 200** will remove the single-tagged VLAN, vlan 200, and associated endpoints. The dual-tagged VLAN, **vlan 200 inner-vlan 300**, will not be deleted.

NOTE
A tagged port can be a part of one or more VLLs, and at the same time be part of one or more regular VLANs and one or more VPLSs as a tagged member.

Viewing the VLL dual-tagged configuration

Use the **show running config** and **show mpls vll** commands to view the VLL dual-tagged configuration. The following shows an example **show running config** output. For examples and details of the **show mpls vll** commands, see "Displaying information about MPLS VLLs" on page 1468.

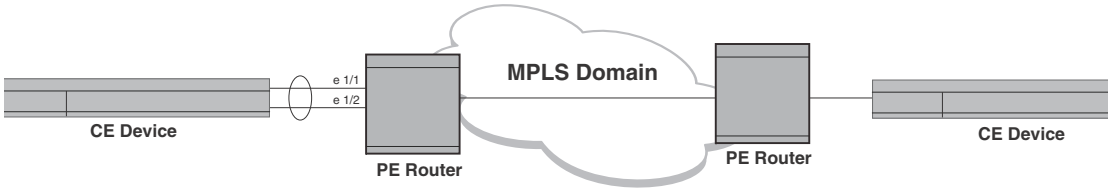
```
NetIron#show run
....
router mpls

vll test 100
  vlan 100 inner-vlan 45
  tag e2
```

Specifying a LAG group as the endpoint of a VLL

The endpoint of a VLL can be a LAG group. When the endpoint of a VLL is a LAG group, the VLL traffic load is distributed to the customer edge (CE) device across all of the LAG group's ports, using a hashing mechanism described in "Hash based load sharing" on page 214.

Figure 192 illustrates a sample configuration where a LAG group of two ports serves as the endpoint of a VLL.

FIGURE 192 Specifying a LAG group as the endpoint of a VLL



To configure a LAG group like the one in Figure 192, enter commands such as the following.

```
NetIron(config)# lag red dynamic
NetIron(config-lag-red)# ports ethernet 1/1 to 1/2
NetIron(config-lag-red)# primary port 1/1
NetIron(config-lag-red)# ports ethernet 1/2
NetIron(config-lag-red)# deploy
```

To configure a VLL like the one in Figure 192, enter commands such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll test2 40000
NetIron(config-mpls-vll)# vll-peer 10.10.10.10
NetIron(config-mpls-vll)# untagged e 1/1
```

NOTES: If you first create a LAG and then configure a VLL instance, the port you specify as the VLL endpoint must also be the port you specified as the primary port of the LAG group:

- If you later delete the LAG from the configuration, only the primary port will still be a port of the VLL and all secondary ports will become normal ports.
- If you specified a tagged endpoint for the VLL instance, all of the ports in the LAG must be tagged.

- Traffic received from any port in the LAG is forwarded to the VLL instance. All traffic is matched to its VLAN.
- Both static and dynamic LAGs are supported.

Enabling VLL MTU enforcement (optional)

You can selectively enable local and remote VLL MTU mismatch checking using the following command.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-mtu-enforcement
```

Syntax: [no] vll-mtu-enforcement

By default, MTU checking is off. You can use the **no** form of the command to disable VLL MTU checking if it is on.

NOTE
You must save the configuration and reload the software for this command to take effect.

Specifying a VLL MTU

Previously, every VLL configured on a NetIron router used the system default max-frame-size as the VLL MTU while establishing the LDP session with its peer. The system default max-frame-size is configured as described in "Setting the maximum frame size globally" on page 681. When this value is changed, the configuration needs to be saved and the router rebooted for the new value to take effect. The change in this value affects all VLLs configured on the router. This parameter is used as the max-frame-size expected on each port of the router and consequently the data plane will not accept or transmit packets larger than this size.

You can now use the **vll-mtu** command that allows you to specify an MTU value per-VLL. If an MTU value is not specified for a VLL, the router continues to use the default max-frame-size for establishing the LDP session with the peer. The MTU value configured per-VLL can be changed dynamically and takes effect immediately. Consequently, the label (if already advertised) is withdrawn from the peer and re-advertised using the new MTU. This occurs irrespective of the state of the VLL. If MTU enforcement checks are enabled and if the MTUs don't match, the VLL stays down with the reason code "MTU mismatch".

NOTE
This parameter is not enforced on the data plane. Consequently, you can still send and receive packets larger than the configured MTU.

To configure a new MTU value for a VLL, use the **vll-mtu** command as shown in the following.

```
NetIron(config-mpls)# vll foundry 40000
NetIron(config-mpls-vll-foundry)# vll-mtu 1000
```

Syntax: [no] vll-mtu <*mtu-value*>

The <*mtu-value*> variable can be set to any value between 64 - 9190.

Generating traps for VLLs

You can enable and disable SNMP traps for VLLs. VLL traps are enabled by default.

To enable VLL traps after they have been disabled, enter the following command.

```
NetIron(config)# snmp-server enable traps mpls vll
```

Syntax: [no] snmp-server enable traps mpls vll

Use the **no** form of the command to disable VLL traps.

MPLS VLL behavior with other features

This section describes the interaction of MPLS VLL with the features sFlow, IFL CAM, and Layer 2 ACLs.

sFlow

sFlow sampling is supported for VLL packets received from customer interfaces. sFlow is not supported for packets received from MPLS uplinks. The following describes the behavior for sFlow for VLL packets received from customer endpoints:

- If the endpoint is untagged, the default VLAN ID in the sFlow sample is used for both the incoming and outgoing VLAN fields in the sFlow sample collection.
- If the endpoint is tagged, the tag is used as both the incoming and outgoing VLAN in the sFlow sample collection.
- If the endpoint is dual-tagged, 4096 is used as the incoming VLAN to indicate that the packet is dual-tagged. If the VLL is in raw-mode, the outer VLAN of the packet is used as the outgoing VLAN ID in the sFlow sample collection. If the VLL is in tagged-mode, the inner VLAN ID of the packet is used as the outgoing VLAN ID in the sFlow sample collection.

  Note that if the endpoint is dual-tagged, the sFlow packet will not contain VLL- or MPLS-specific information.

Table 241 illustrates the above points.

TABLE 241 Source and destination VLAN in an sFlow sampled VLL packet

Endpoint	Source VLAN	Destination VLAN
Untagged	Default VLAN	Default VLAN
Single-tagged	Incoming VLAN	Incoming VLAN
Dual-tagged	4096	Raw mode: Incoming outer VLAN Tagged mode: Incoming inner VLAN

IFL CAM

For dual-tagged VLL instances, IFL CAM entries are used for the service lookup. The default system value for IFL CAM is 8192, which can be modified up to a maximum of 81920 entries using the CLI command **system-max ifl-cam** <*number*>. For more information about modifying the number of IFL CAM entries supported, refer to "Configuring system max values" on page 106.

Layer 2 ACLs

When the port and VLAN combination of a Layer 2 ACL matches with any VLL endpoint, the ACL is applied. For dual-tagged VLL endpoints, the Layer 2 ACL is applied based on the port and outer VLAN combination, if it is configured.

Displaying MPLS VLL information

You can display the following information about the MPLS VLL configuration on the device:

- Information about individual MPLS VLLs configured on the device
- Information about detailed MPLS VLLs configured on the device
- Information about LDP sessions between VLL peers
- Information about VLL Endpoint Statistics
- Information about packets sent between VLL endpoints and MPLS uplinks

Displaying information about MPLS VLLs

Use the following command to display information about MPLS VLLs.

```
NetIron# show mpls vll
Name    VC-ID   Vll-peer      End-point                          State  Tunnel-LSP
test    10      11.11.11.11   tag vlan 2    e 1/10               UP     tn17
test2   100     --            tag vlan 100 inner-vlan 45  e2/1   DOWN
```

Syntax: show mpls vll

NOTE
Show commands have been enhanced to include the full MPLS name. Previously, the MPLS name was truncated because it exceeded the character length. Now, the MPLS name is text wrapped to display the full name.

For each MPLS VLL on the device, the following information is displayed.

TABLE 242 Output from the show mpls vll command

This field...	Displays...
Name	The configured name of the VLL.
VC-ID	The user-configurable ID as defined in draft-ietf-pwe3-control-protocol-14.txt.
Vll-peer	The remote PE router. This should be the same as the LSP destination for the LSPs that the VLL is transported over.

TABLE 242 Output from the show mpls vll command (Continued)

This field...	Displays...
End-point	How packets are forwarded once they reach the egress LER. This can be one of the following • "untagged <*portnum*>" – Forward the packet out the specified port as untagged. • "tag VLAN <*vlan-id*> <*portnum*>" – Tag the packet with the specified VLAN ID and forward the packet out the specified port. • tag VLAN <*vlan-id*> inner-vlan <*vlan-id*> – Tag the packet with the specified outer and inner VLAN IDs and forward the packet out the specified port. • "undefined" – An endpoint has not been configured for this VLL
State	The current state of the VLL. This can be either UP or DOWN. Data can be forwarded over the VLL only when the state is UP.
Tunnel-LSP	The name of the RSVP-signalled LSP that has been selected to carry the VLL traffic through the MPLS domain

Displaying detailed information about MPLS VLLs

NOTE
The display changes to the **show mpls vll detail** command output are supported on .

To display detailed information about the VLLs configured on the device, use the **show mpls vll detail** command. The **show mpls vll detail** command has changed. The following DOWN states (with their respective reasons), are introduced in the output of the **show mpls vll detail** command. For more information on the DOWN states, refer to Table 243 on page 1472.

- The state, DOWN - PW is Down (Reason: Out of VC labels)
- The state, DOWN - PW is Down (Reason: LDP session is down)

NOTE: The state, DOWN - no LDP session to vll-peer is removed from the output, and replaced with the state, DOWN - PW is Down (Reason: LDP session is down).

- The state, DOWN - PW is Down (Reason: Out of Memory)
- The state, DOWN - PW is Down (Reason: Waiting for Remote VC label)
- The state, DOWN - PW is Down (Reason: MTU mismatch Local- MTU <mtu-value>, Remote-MTU <mtu-value>)
- The state, DOWN - PW is Down (Reason: VC type mismatch, Local VC type: <vc-type>, Remote VC type: <vc-type>)

NOTE: The state, DOWN - VC Type Mismatch in VC signalling, Local VC type <vc-type>, Remote VC type <vc-type>is removed from the output, and replaced with the state, DOWN - PW is Down (Reason: VC type mismatch, Local VC type:<vc-type>, Remote VC type: <vc-type>).

For example, the state, DOWN - Pseudo Wire (PW) is Down (Reason: MTU mismatch Local-MTU 1500, Remote -MTU 1400) indicates that PW is down because the MTU values between the local MTU and the remote MTU are not equal, as shown in the example below.

```
NetIron# show mpls vll detail
VLL VLL1 VC-ID 1001
State:     DOWN -PW is Down (Reason: MTU mismatch Local-MTU 1500, Remote-MTU
1400)
End-point: tagged vlan 1001 e 2/20
IFL-ID:    --
Vll-peer:  21.21.21.21
Local VC type:   tag                Remote VC type:   --
Local label:                        Remote label:     --
Local group-id:  0                  Remote group-id:  --
Local VC MTU:   1500                Remote VC MTU:   1400
COS:             --                 Tunnel LSP:      LSP_MLXe21 (tnl0)
```

The example below displays state, DOWN - PW is Down (Reason: LDP session is down) for a single-tagged VLL instance.

```
NetIron# show mpls vll detail
VLL VLL1 VC-ID 1001
State:     DOWN -PW is Down (Reason: LDP session is down)
End-point: tagged vlan 1001 e 2/20
IFL-ID:    --
Vll-peer:  21.21.21.21
Local VC type:    tag                 Remote VC type:    --
Local label:      --                  Remote label:      --
Local group-id:    0                  Remote group-id:   --
Local VC MTU:    1500                 Remote VC MTU:     --
COS:              --                  Tunnel LSP:        LSP_MLXe21 (tnl0)
```

The UP state is introduced in the output of the **show mpls vll detail** command. The UP state indicates that VLL is operational and packets are able to flow. The following example displays an output of a dual-tagged VLL instance in an UP state.

```
NetIron# show mpls vll detail
VLL VLL1 VC-ID 1001
State:      UP
End-point: tagged vlan 1001 e 2/20
IFL-ID:           --
Vll-peer:  21.21.21.21
Local VC type:     tag                    Remote VC type:   tag
Local label:      800001                  Remote label:     800002
Local group-id:    0                      Remote group-id:  0
Local VC MTU:     1500                    Remote VC MTU:    1500
COS:               --                     Tunnel LSP:       LSP_MLXe21 (tnl0)
```

The DOWN states, Waiting for PW Up, and Waiting for VC Withdrawal Completion is introduced in the output of the **show mpls vll detail** command, and are described in more detail below.

- The state, DOWN - Waiting for PW Up indicates that VLL is waiting for MPLS to bring up the session.
- The state, DOWN - Waiting for VC Withdrawal Completion indicates that PW is down, and VLL is waiting for MPLS to withdraw the labels that VLL has requested.

The following example below displays the state, Down - Waiting for PW Up.

```
NetIron# show mpls vll detail
VLL VLL1 VC-ID 1001
State:     DOWN -Waiting for PW Up
End-point: tagged vlan 1001 e 2/20
IFL-ID:    --
Vll-peer:  21.21.21.21
Local VC type:   tag            Remote VC type:   --
Local label:     --             Remote label:     --
Local group-id:  0              Remote group-id:  --
Local VC MTU:   1500            Remote VC MTU:    --
COS:            --              Tunnel LSP:       LSP_MLXe21 (tnl0)
```

Syntax: show mpls vll detail | *<vll-name>*

For each configured VLL, the command displays the following information in Table 263.

TABLE 243 Output from the show mpls vll detail command

This field...	Displays...
state	The current state of the VLL. This can be one of the following • "UP" VLL is operational - packets can flow. • "DOWN - configuration incomplete" A required configuration statement is missing. • "DOWN - endpoint port to CE is down" The physical endpoint port that should be connected to the Customer Edge device is down due to a link outage or is administratively disabled. • "DOWN - no tunnel LSP to vll-peer" Cannot find a working LSP. • "DOWN - PW is Down (Reason: LDP session is down)" LDP session is not yet ready. • "DOWN - Waiting for PW Up" VLL is waiting for MPLS to bring up the session. • "DOWN - Waiting for VC withdrawal Completion" PW is down, and VLL is waiting for MPLS to withdraw the labels that VLL has requested. • "DOWN - PW is Down (Reason: Out of VC labels)" PW is down; VC labels are not available. • "DOWN - PW is Down (Reason: Out of Memory)" PW is down; there is not sufficient memory available. • "DOWN - PW is Down (Reason: Waiting for Remote VC label)" PW is down; waiting for remote peer's VC label to be advertised. • "DOWN - waiting for VC label binding from vll-peer" The device has advertised its VC label binding to the VLL peer, but has not yet received the peer's VC label binding. • "DOWN - PW is Down (Reason: MTU mismatch Local- MTU <mtu-value>, Remote-MTU <mtu-value>)" PW is down and the MTU values for the local and remote peers are not equal. • "DOWN - PW is Down (Reason: VC type mismatch, Local VC type: <*vc-type*>, Remote VC type: <*vc-type*>" – The session cannot be brought up because the VC types of the local and remote peers are not equal. The possible values for the <*vc-type*> variable are 5 for raw mode or 4 for tagged mode.
End-point	How packets are forwarded once they reach the egress LER. This can be one of the following: • "untagged <*portnum*>" - Forward the packet out the specified port as untagged. • "tagged VLAN <*vlan_id*> <*portnum*>" - Tag the packet with the specified VLAN ID and forward the packet out the specified port. • "tagged VLAN <*vlan-id*> inner-vlan <*vlan-id*>" – Tag the packet with the specified outer and inner VLAN IDs and forward the packet out the specified port. • "undefined" - An endpoint has not been configured for this VLL.
IFL-ID	The Internal Forwarding Lookup Indentifier (IFL-ID) that is allocated to each Local VLL instance that has at least one dual-tagged endpoint. For instances that do not have dual-tagged endpoints, the IFL-ID is displayed as "–".
Vll-peer	The remote PE router. This should be the same as the LSP destination for the LSPs that the VLL is transported over.
Local VC Type	Indicates whether the local VC is in Raw-mode or Tagged-mode.
Remote VC Type	Indicates whether the remote VC is in Raw-mode or Tagged-mode.
Local Label	The VC label value locally allocated for this VLL. Packets forwarded from the VLL peer to this device are expected to contain this label. This is the label that is advertised to the VLL peer through LDP.

TABLE 243 Output from the show mpls vll detail command (Continued)

This field...	Displays...
Remote Label	The VC label allocated by the VLL peer and advertised to this device through LDP. The device applies this label to outbound MPLS packets sent to the VLL peer.
Local Group-id	The VLL group-ID (defined in draft-martini-l2circuit-trans-mpls-07.txt) advertised to the VLL peer through LDP. In this release, this is always zero.
Remote Group-id	The VLL group-ID selected and advertised by the VLL Peer.
Local VC MTU	The MTU value configured for this local VC.
Remote VC MTU	The MTU value advertised from the VLL peer.
COS	The optional COS setting for the VLL. If a COS value is set, the device will attempt to select a tunnel LSP that also has this COS value. The COS value can be between 0 - 7.
Tunnel LSP	The name, as well as internal tunnel index number, of the tunnel LSP selected for the VLL.

In a situation when MPLS may run out local resources, an error state is displayed in the **show mpls vll detail** command output. The errors states, DOWN - VC binding Failed, and DOWN - VC withdrawal Failed are displayed in the examples below. To recover from this state, the user is required to delete the failed peer and reconfigure it.

```
NetIron# show mpls vll detail
VLL VLL1 VC-ID 1001
State:     DOWN -VC binding Failed
End-point: tagged vlan 1001 e 2/20
IFL-ID:    --
Vll-peer:  21.21.21.21
Local VC type:  tag                Remote VC type:   --
Local label:    --                 Remote label:     --
Local group-id: 0                  Remote group-id:  --
Local VC MTU:  1500                Remote VC MTU:    --
COS:           --                  Tunnel LSP:       LSP_MLXe21 (tnl0)
```

The example below displays the state, DOWN - VC withdrawal failed.

```
NetIron# show mpls vll detail
VLL VLL1 VC-ID 1001
State:     DOWN -VC withdrawal Failed
End-point: tagged vlan 1001 e 2/20
IFL-ID:    --
Vll-peer: 21.21.21.21
Local VC type:   tag              Remote VC type:   --
Local label:     --               Remote label:     --
Local group-id:  0                Remote group-id:  --
Local VC MTU:   1500              Remote VC MTU:    --
COS:            --                Tunnel LSP:       LSP_MLXe21 (tnl0)
```

VLL will generate the following warning messages. MPLS will also generate a warning message, and it will be displayed on the console as shown in the example below.

```
WARNING: VLL Id 1 with Peer 1.1.1.1 is placed in VC Bind failure state due to MPLS
resource failure
```

```
WARNING: VLL Id 2 with Peer 2.2.2.2 is placed in VC Withdraw failure state due to
MPLS resource failure
```

Displaying LDP information

To display information about the state of the LDP connection between the device and VLL peers, enter the following command.

```
NetIron# show mpls ldp peer
Peer LDP ID State Num-VLL Num-VPLS-Peer
11.11.11.11:0 Operational 32008 0
13.13.13.13:0 Operational 0 0
```

Syntax: show mpls ldp peer

For each VLL peer, the command displays the following information:

TABLE 244 Output from the show mpls ldp target-peer command

This field...	Displays...
Peer-addr	The IP addresses of VLL peers.
State	The state of the LDP session with the VLL peer. This can be one of the following "Unknown" LDP session establishment has not started for this peer, normally because no Hello messages have been received from the peer. In this situation, that peer will not show up in the output of the **show mpls ldp session** command. "Nonexistent", "Initialized", "OpenRec", "OpenSent", or "Operational" LDP session states, as defined in RFC 3036.

To display information about LDP sessions between the device and VLL peers.

```
NetIron# show mpls ldp session
Peer LDP Ident: 192.168.2.100:1, Local LDP Ident: 11.1.1.1:1
  Active: no, State: Operational
  TCP connection: 11.1.1.1:646--22.2.2.2:9001, State: ESTABLISHED
  Addresses bound to peer LDP Ident:
    1.1.1.2
    10.1.1.2
    20.1.1.2
    22.2.2.2
```

Syntax: show mpls ldp session [*<label-space-id>* **| detail | brief]**

For each established LDP session, the command displays the following information.

TABLE 245 Output from the show mpls ldp session command

This field...	Displays...
Peer LDP Ident	The VLL peer's LDP identifier, consisting of the LSR ID and label space ID.
Local LDP Ident	The device's LDP identifier.
Active	Whether this LSR is playing an active role in session establishment.
State	The LDP session state, as defined in RFC 3036. This can be "Nonexistent", "Initialized", "OpenRec", "OpenSent", or "Operational".

TABLE 245 Output from the show mpls ldp session command (Continued)

This field...	Displays...
TCP connection, state	The TCP local or remote IP address, port and state.
Addresses bound to peer LDP Ident	IP addresses carried in the VLL peer's LDP Address messages.

To display information about LDP sessions between a specified router and VLL peers.

```
NetIron# #show mpls ldp session 22.22.22.22
Peer LDP ID: 22.22.22.22:0, Local LDP ID: 24.24.24.24:0, State: Operational
  Adj: Link, Role: Active, Next keepalive: 0 sec, Hold time left: 30 sec
  Keepalive interval: 6 sec, Max hold time: 36 sec
  Neighboring interfaces: e1/4
  TCP connection: 24.24.24.24:9012--22.22.22.22:646, State: ESTABLISHED
  Next-hop addresses received from the peer:
    22.22.22.22  40.40.40.1  10.10.10.2
```

The command displays the following information.

TABLE 246 Output from the show mpls ldp session command

This field...	Displays...
Peer LDP Ident	The VLL peer's LDP identifier, consisting of the LSR ID and label space ID.
Local LDP Ident	The device's LDP identifier.
State	The LDP session state, as defined in RFC 3036. This can be "Nonexistent", "Initialized", "OpenRec", "OpenSent", or "Operational".
Adj	The type of adjacency formed with a peer. Possible values are Link, or Targeted.
Role	This can be either of the following values Active or Passive.
Next keepalive	The number of seconds after which a "Hello" message is sent to a peer.
Hold time left	The number of seconds after which a session can be terminated if a "Hello" message is not received from the peer within this time.
Keepalive interval	The frequency within which LDP "Hello" messages are sent out.
Max hold time	The length of time the device waits for a "Hello" message from its peer before terminating the session.
Neighboring interfaces	The physical interfaces on which the adjacency to the neighbor is formed.
TCP connection, state	The TCP local or remote IP address, port and state.
Next-hop addresses received from the peer	IP addresses carried in the VLL peer's LDP Address messages.

Displaying VLL endpoint statistics

You can display VLL Endpoint traffic statistics to see the forwarding counters for each VLL configured on the system. The display is shown so that, for a given port range that receives traffic, it shows the number of packets arriving from the customer endpoint and the number of packets arriving from the MPLS core and going to the customer interface.

To display all VLL traffic statistics on a NetIron router, enter the following command.

```
NetIron# show mpls statistics vll
VLL-Name       VLL Port(s)    VLL-Ingress-Pkts  VLL-Egress-Pkts
---------      ----------     --------------    ------------
VLL1           e1/1           100               100
VLL2           e1/4           100               100
```

NOTE
The VLL name is repeated for each module from where the statistics are collected, to be displayed on the Management console.

To display VLL traffic statistics for a VLL instance specified by its VLL name, enter the following command.

```
NetIron# show mpls statistics vll vll1
VLL-Name       VLL Port(s)    VLL-Ingress-Pkts  VLL-Egress-Pkts
---------      ----------     --------------    ------------
VLL1           e1/1           100               100
```

To display VLL traffic statistics for a VLL instance specified by its VLL ID, enter the following command.

```
NetIron# show mpls statistics vll 4
VLL-Name       VLL Port(s)    VLL-Ingress-Pkts  VLL-Egress-Pkts
---------      ----------     --------------    ------------
VLL1           e1/1           100               100
```

Syntax: show mpls statistics vll [<*vll-name*> | <*vll-id*>]

The <*vll-name*> variable is the configured name for a VLL instance.
The <*vll-id*> variable is the ID of a VLL instance.

For following information is displayed in the **show mpls statistics vll** command.

TABLE 247 Output from the **show mpls vll command**

This field...	Displays...
VLL-Name	The configured name of the VLL instance.
VLL-Ports	The port where the traffic is monitored.
VLL-Ingress-Pkts	Packets arriving from the Customer Endpoint.
VLL-Egress-Pkts	Packets arriving from the MPLS core and going to the customer interface

Clearing VLL traffic statistics

To clear the entries stored for all VLL statistics, enter a command such as the following.

```
NetIron# clear mpls statistics vll
```

Syntax: clear mpls statistics vll [<*vll-name*> | <*vll-id*>]

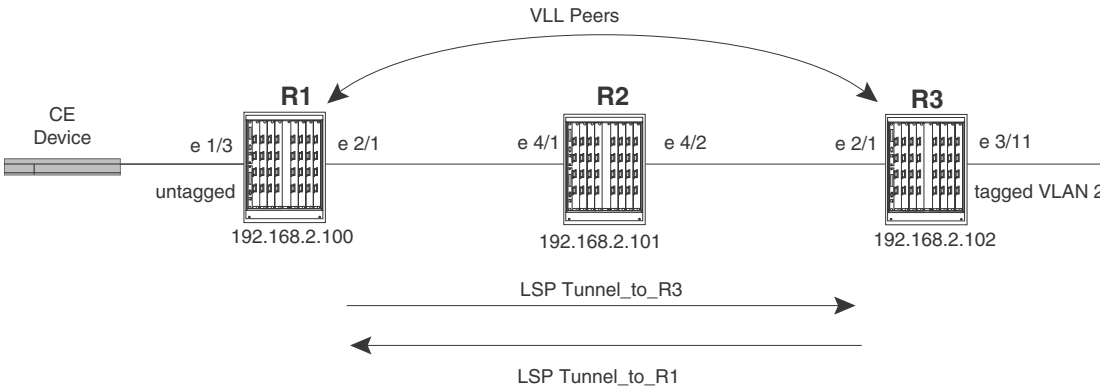
The <*vll-name*> variable is the configured name for a VLL instance.

The <*vll-id*> variable is the ID of a VLL instance.

Sample MPLS VLL configuration

Figure 193 depicts a sample VLL configuration.

FIGURE 193 MPLS VLL configuration



In this example, routers R1 and R3 are Provider Edge (PE) routers configured as VLL peers. R1 and R3 have established an LDP session to exchange VLL label information. When the LDP session is established, each router advertises its locally assigned VC label and VC ID to its VLL peer.

RSVP-signalled (tunnel) LSPs have been established in each direction between the two routers. When the CE device forwards a Layer 2 packet to R1, the router assigns the packet to an RSVP-signalled LSP whose destination is R3. R1 encapsulates the packet as an MPLS packet, adding a tunnel label and the VC label advertised to the router by R3. The MPLS packet is then forwarded over the outbound interface indicated by the tunnel label to the next hop in the LSP.

When the MPLS packet reaches R2, the penultimate LSR in the tunnel LSP, R2 pops the tunnel label, leaving the packet with only the VC label, then forwards the packet to R3.

R3 examines the VC label in the packet. On R3, the VC label is mapped to the user-specified endpoint for the VLL. In this example, the endpoint consists of VLAN ID 200 and interface 3/11. R3 then pops the VC label, tags the Layer 2 packet with VLAN 200, then forwards the packet out interface 3/11.

In the opposite direction, R3 assigns traffic received from the CE device to an RSVP-signalled LSP destined for R1, pushes tunnel and VC labels onto the packets, and forwards them to the next hop in LSP. When the packets reach R1, the router pops the VC label and forwards the Layer 2 packets out the interface indicated by the VLL endpoint. In this example, the endpoint consists of interface 1/3, so the packets are forwarded untagged out interface 1/3 to the CE device.

Router R1

The following commands configure Router R1 in Figure 193.

```
NetIron(config)# ip router-id 192.168.2.100
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 0
NetIron(config-ospf-router)# exit
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface e 2/1
```

```
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# traffic-engineering ospf
NetIron(config-mpls-policy)# exit
NetIron(config-mpls)# lsp Tunnel_To_R3
NetIron(config-mpls-lsp)# to 192.168.2.102
NetIron(config-mpls-lsp)# enable
NetIron(config-mpls-lsp)# exit
NetIron(config-mpls)# vll VLL_to_R3 40000
NetIron(config-mpls-vll)# vll-peer 192.168.2.102
NetIron(config-mpls-vll)# untagged e 1/3
NetIron(config-mpls-vll)# exit
NetIron(config)# interface loopback 1
NetIron(config-lbif-1)# port-name Generic All-Purpose Loopback
NetIron(config-lbif-1)# ip address 192.168.2.100/32
NetIron(config-lbif-1)# ip ospf area 0
NetIron(config-lbif-1)# exit
NetIron(config)# interface e 1/3
NetIron(config-if-e100-1/3)# port-name VLL_endpoint
NetIron(config-if-e100-1/3)# enable
NetIron(config-if-e100-1/3)# exit
NetIron(config)# interface e 2/1
NetIron(config-if-e1000-2/1)# port-name Connection_to_R2
NetIron(config-if-e1000-2/1)# enable
NetIron(config-if-e1000-2/1)# ip address 192.168.37.1/30
NetIron(config-if-e1000-2/1)# ip ospf area 0
NetIron(config-if-e1000-2/1)# exit
```

Router R2

The following commands configure Router R2 in Figure 193.

```
NetIron(config)# ip router-id 192.168.2.101
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 0
NetIron(config-ospf-router)# exit
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface e 4/1 to e 4/2
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# traffic-engineering ospf
NetIron(config-mpls-policy)# exit
NetIron(config)# interface e 4/1
NetIron(config-if-e1000-4/1)# enable
NetIron(config-if-e1000-4/1)# ip address 192.168.37.2/30
NetIron(config-if-e1000-4/1)# ip ospf area 0
NetIron(config-if-e1000-4/1)# exit
NetIron(config)# interface e 4/2
NetIron(config-if-e1000-4/2)# enable
NetIron(config-if-e1000-4/2)# ip address 192.168.41.2/30
NetIron(config-if-e1000-4/2)# ip ospf area 0
NetIron(config-if-e1000-4/2)# exit
NetIron(config)# interface loopback 1
NetIron(config-lbif-1)# port-name Generic All-Purpose Loopback
NetIron(config-lbif-1)# ip address 192.168.2.101/32
NetIron(config-lbif-1)# ip ospf area 0
NetIron(config-lbif-1)# exit
```

Router R3

The following commands configure Router R3 in Figure 193.

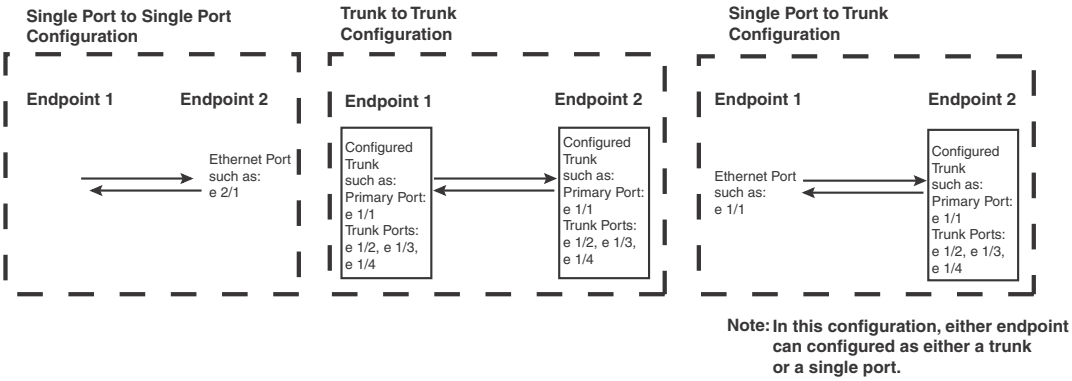
```
NetIron(config)# ip router-id 192.168.2.102
NetIron(config)# router ospf
NetIron(config-ospf-router)# area 0
NetIron(config-ospf-router)# exit
NetIron(config)# router mpls
NetIron(config-mpls)# mpls-interface e 2/1
NetIron(config-mpls)# policy
NetIron(config-mpls-policy)# traffic-engineering ospf
NetIron(config-mpls-policy)# exit
NetIron(config-mpls)# lsp Tunnel_To_R1
NetIron(config-mpls-lsp)# to 192.168.2.100
NetIron(config-mpls-lsp)# enable
NetIron(config-mpls-lsp)# exit
NetIron(config-mpls)# vll VLL_to_R1 40000
NetIron(config-mpls-vll)# vll-peer 192.168.2.100
NetIron(config-mpls-vll)# vlan 200
NetIron(config-mpls-vll-vlan)# tagged e 3/11
NetIron(config-mpls-vll-vlan)# exit
NetIron(config-mpls-vll)# exit
NetIron(config)# interface loopback 1
NetIron(config-lbif-1)# port-name Generic All-Purpose Loopback
NetIron(config-lbif-1)# ip address 192.168.2.102/32
NetIron(config-lbif-1)# ip ospf area 0
NetIron(config-lbif-1)# exit
NetIron(config)# interface e 3/11
NetIron(config-if-e100-3/11)# port-name VLL_endpoint
NetIron(config-if-e100-3/11)# enable
NetIron(config-if-e100-3/11)# exit
NetIron(config)# interface e 2/1
NetIron(config-if-e1000-2/1)# port-name Connection_to_R2
NetIron(config-if-e1000-2/1)# enable
NetIron(config-if-e1000-2/1)# ip address 192.168.41.1/30
NetIron(config-if-e1000-2/1)# ip ospf area 0
NetIron(config-if-e1000-2/1)# exit
```

Local VLL

Local VLL is used to create a Virtual Leased Line (VLL) circuit with endpoints in the same NetIron router. A Local VLL can be configured between two ports in a router, two LAGs in a router or between a port and a LAG as shown in Figure 194. Each entity (port or LAG) is identified as either “Endpoint 1” or Endpoint 2”.

NOTE
LAGs supported include server LAGs and per-packet server LAGs. LACP LAGs are not supported.

FIGURE 194 Local VLL port and LAG configurations



NOTE
When configuring a LAG as an endpoint, only the primary port of the LAG is specified in the Local VLL configuration.

NOTE
Packets that arrive on an interface with the same destination MAC address as the interface are forwarded in hardware just like packets with other destination addresses.

The endpoints connected to the Local VLL can be untagged or tagged as members of the same or different VLANs. Using this function of Local VLL, a router can receive packets with a particular tag or no tag on one endpoint and forward them to the Local VLL's other endpoint which may be untagged or tagged with a different VLAN tag. When so configured, the tags within the packets are changed to reflect the configuration of the egress port as they leave the router.

Local VLL configuration examples

Local VLL supports traffic flows between any combination of single-tagged, untagged, and dual-tagged ports. Some scenarios are described and illustrated in the following configuration examples.

Example of a Local VLL configured for single-tagged VLAN traffic on both ports

In Figure 195 the Local VLL named "Test1" contains Ethernet ports 1/1 and 2/1. Port 1/1 is a member of VLAN 100 and port 2/1 is a member of VLAN 200. Because both ports belong to Local VLL "Test1" traffic tagged with VLAN 100 will be able to reach nodes within VLAN 200 and traffic tagged with VLAN 200 will be able to reach nodes within VLAN 100. Traffic that ingresses on port 1/1 must have a tag with the value "100" and will egress on port 2/1 with a tag value of "200". Traffic that ingresses on port 2/1 must have a tag with the value "200" and will egress on port 2/1 with a tag value of "100".

FIGURE 195 Local VLL "Test1" with 2 tagged VLANs

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test1
NetIron(config-mpls-vll-lo-test1)# vlan 100
NetIron(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/1
NetIron(config-mpls-vll-lo-test1-vlan)# vlan 200
NetIron(config-mpls-vll-lo-test1-vlan)# tagged ethernet 2/1
```

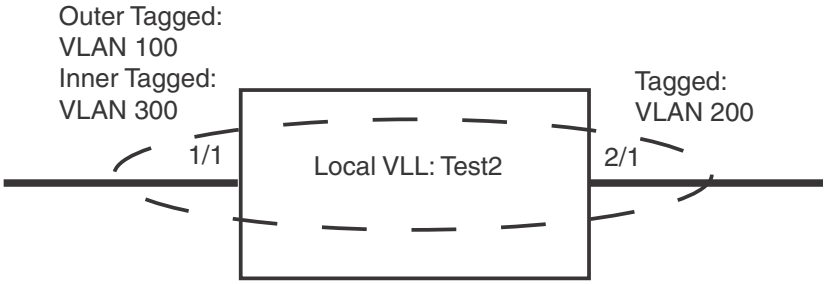
Example of a Local VLL configured for dual-tagged and single-tagged VLAN traffic

You can configure a Local VLL with ports that are configured for dual tags. In a dual-tagged configuration, the packets contain an outer tag and an inner tag. One or both of the VLANs configured for the Local VLL will have an inner VLAN configured in addition to the default outer VLAN.

Under this configuration, a router can receive packets with a two tags on one endpoint and forward them to the Local VLLs other endpoint either untagged, tagged with a single tag or tagged with an inner and and outer VLAN tag. Where dual-tagging is used within a Local VLL, the system allocates an Internal Lookup Indentifier (IFL-ID) for the Local VLL instance.

In Figure 196 the Local VLL named "Test2" contains Ethernet ports 1/1 and 2/1. Port 1/1 is a member of VLAN 100 and is configured to accept packets with an inner VLAN tag value of 300. Port 2/1 is a member of VLAN 200. Because both ports belong to Local VLL "Test2" traffic tagged with outer VLAN tag 100 and inner VLAN tag 300 will be able to reach nodes within VLAN 200 and traffic tagged with VLAN 200 will be able to reach nodes within VLAN 100. Traffic that ingresses on port 1/1 must have an outer tag with the value "100" and an inner tag with the value "300" and will egress on port 2/1 with a tag value of "200". Traffic that ingresses on port 2/1 must have a tag with the value "200" and will egress on port 1/1 with an inner tag value of "100" and an outer tag value of "300".

FIGURE 196 Local VLL "Test2" with 1 single-tagged VLAN and 1 dual-tagged VLAN



```
NetIron(config)# system-max ifl-cam 16384
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test2
NetIron(config-mpls-vll-lo-test1)# vlan 100 inner-vlan 300
NetIron(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/1
NetIron(config-mpls-vll-lo-test1-vlan)# vlan 200
NetIron(config-mpls-vll-lo-test1-vlan)# tagged ethernet 2/1
```

A shown in the following example, you can use the **show mpls vll-local detail** command to see that an IFL-ID has been created for this Local VLL instance.

```
NetIron#show mpls vll-local detail
VLL test2 VLL-ID 1 IFL-ID 4096
  State: UP
  End-point 1:      tagged  vlan 100   inner-vlan 300    e 1/1
                    COS: --
  End-point 2:      tagged  vlan 200   e 2/1
                    COS: --
```

Example of a Local VLL configured for single-tagged and untagged VLAN traffic

In Figure 197 the Local VLL named “Test3” contains Ethernet ports 1/1 and 2/1. Port 1/1 is a member of VLAN 100 and port 2/1 is untagged. Because both ports belong to Local VLL “Test3, traffic tagged with VLAN 100 will be able to reach nodes attached to the untagged port and traffic from the untagged port will be able to reach nodes within VLAN 100.

FIGURE 197 Local VLL “Test3” with 1 tagged VLAN and 1 untagged port

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test3
NetIron(config-mpls)# untagged ethernet 2/1
NetIron(config-mpls-vll-lo-test1)# vlan 100
NetIron(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/1
```

Local VLL QoS

You can optionally specify Class of Service (COS) on a per-endpoint (EP) basis. This COS value applies to inbound traffic on the endpoint. If a COS value is not specified, the port’s configured priority and the packet’s 802.1p priority are used to determine the internal priority, as described for the following traffic flows.

Untagged Endpoint 1 (EP1) to Untagged Endpoint 2 (EP2).

1. If available, use the configured COS value on untagged EP1 otherwise go to step 2.

2. If there is a configured port priority on untagged EP1 use that priority; otherwise go to step 3.
3. If there is neither a COS value or priority configured (as described in steps 1 and 2), the default “best effort” priority is used.

Tagged Endpoint 1 (EP1) to Tagged Endpoint 2 (EP2) .

1. If available, use the configured COS value on tagged EP1 otherwise go to step 2.
2. The 802.1p priority of incoming packets on the tagged EP1 is merged with the port priority of EP1.

The 802.1p priority for packets outbound from EP2 is determined as described in 9, “Configuring Quality of Service for the NetIron MLX”.

Untagged Endpoint 1 (EP1) to Tagged Endpoint 2 (EP2).

1. If available, use the configured COS value on untagged EP1 otherwise go to step 2.
2. If there is a configured port priority on untagged EP1 use that priority; otherwise go to step 3.
3. If there is neither a COS value or priority configured (as described in steps 1 and 2), the default “best effort” priority is used.

The 802.1p priority for packets outbound from EP2 is determined as described in 9, “Configuring Quality of Service for the NetIron MLX”.

Tagged Endpoint 1 (EP1) to Untagged Endpoint 2 (EP2).

1. If available, use the configured COS value on tagged EP1 otherwise go to step 2.
2. The 802.1p priority of incoming packets on the tagged EP1 is merged with the port priority of EP1.

Double-Tagged Endpoint 1 (EP1) to Double-Tagged Endpoint 2 (EP2).

1. The internal priority is determined as described in 9, “Configuring Quality of Service for the NetIron MLX”.
2. If Vll-local COS is configured, this value overrides internal priority.
3. By default, the outgoing outer VLAN COS is mapped from the internal priority by use of the egress encoding map. The internal priority does not affect the outgoing inner VLAN COS.
4. The outgoing outer VLAN COS is preserved if “qos pcp encode-policy off” is configured on the outgoing interface.
5. The outgoing inner VLAN COS is preserved—it is the same as the incoming packet’s inner VLAN COS.

Single-Tagged Endpoint 1 (EP1) to Double-Tagged Endpoint 2 (EP2).

1. The internal priority is determined as described in 9, “Configuring Quality of Service for the NetIron MLX”.
2. If Vll-local COS is configured, this value overrides internal priority.
3. The outgoing outer VLAN COS is mapped from internal priority through use of the egress encoding map by default. The internal priority does not affect the outgoing inner VLAN COS.

4. The outgoing outer VLAN COS is preserved if “qos pcp encode-policy off” is configured on the outgoing interface.
5. The outgoing inner VLAN COS is the COS in the incoming packet.

Untagged Endpoint 1 (EP1) to Double-Tagged Endpoint 2 (EP2).

1. The internal priority is determined as described in 9, “Configuring Quality of Service for the NetIron MLX”.
2. If Vll-local COS is configured, this value overrides internal priority.
3. The outgoing outer VLAN COS is mapped from internal priority through the use of the egress encoding map by default. The internal priority does not affect the outgoing inner VLAN COS.
4. The outgoing outer VLAN COS is 0 if “qos pcp encode-policy off” is configured on the outgoing interface.
5. The outgoing inner VLAN COS would be 0.

Double-Tagged Endpoint 1 (EP1) to Single-Tagged Endpoint 2 (EP2).

1. The internal priority is determined as described in 9, “Configuring Quality of Service for the NetIron MLX”.
2. If Vll-local COS is configured, its value overrides internal priority.
3. By default, the outgoing VLAN COS is mapped from the internal priority by use of the egress encoding map.
4. The outgoing VLAN COS is equal to the incoming, inner VLAN COS if “qos pcp encode-policy off” is configured on the outgoing interface.

CoS behavior for Local VLL

NOTE
This section assumes that you understand how QoS works. For details, see For details, see 9, “Configuring Quality of Service for the NetIron MLX”.

Table 248 describes the expected Class of Service (CoS) behavior for VLL packets when Local VLL is in effect.

TABLE 248 Expected class of service behavior for Local VLL

Local VLL endpoints	Incoming packet		Outgoing packet	
	Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN
Dual-tagged to dual-tagged	X	Y	X’ or X	Y
Single-tagged to dual-tagged	X	N/A	X’ or X	X
Untagged to dual-tagged	N/A	N/A	X’ or 0	0
Dual-tagged to single-tagged	X	Y	X’ or Y	N/A

Legend for Table 248

X = original outer VLAN CoS

Y = original inner VLAN CoS

X' = mapped CoS from internal priority (X contributes to internal priority) using CoS encode table

Configuring Local VLL

Configuring Local VLL uses the following procedures:

- "Local VLL configuration"
- "Specifying Local VLL endpoints"
- "Configuring Local VLL QoS (optional)"

Local VLL configuration

Local VLL is configured under router mpls as shown.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test1
```

Syntax: [no] vll-local <*vll-name*>

Specifying Local VLL endpoints

Local VLL can be configured between any combination of untagged, single-tagged, and dual-tagged endpoints.

The following sections describe how to configure VLL Endpoints:

Configuring an untagged endpoint

To configure untagged port 1/1 into Local VLL instance "test1" use the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test1
NetIron(config-mpls-vll-test1)# untagged ethernet 1/1
```

Syntax: [no] untagged ethernet <*slot/port*>

Configuring a single-tagged endpoint

Tagged ports are configured under a VLAN ID. This ID is only meaningful for the tagged port.

For tagged ports, a <*vlan-id, port*> pair constitutes a VLL endpoint. if a port is currently a member of a non-default VLAN as an untagged port, it must be returned to the default VLAN before it can be assigned to a VLL as a tagged port.

To configure tagged port 1/2 with VLAN 200 into Local VLL instance "test1" use the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test1
NetIron(config-mpls-vll-lo-test1)# vlan 200
NetIron(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/2
```

Syntax: vlan <*VLAN-ID*>

The range for **VLAN ID** is 1 – 4094. (This parameter range excludes the default VLAN ID.)

Syntax: [no] tagged ethernet <*slot/port*>

Configuring a dual-tagged endpoint

Dual-tagged ports are configured under a VLAN ID. The main difference between single and dual tagged configuration is that in the dual-tagged configuration, a parameter for **inner-vlan** is configured.

Considerations when configuring the Local VLL with dual tagged endpoints

Before configuring a Local VLL to operate with Dual Tagged Endpoints, you must consider the following:

- The System Max value for IFL CAM must be changed from its default value of 0 (which does not support this feature) to a higher value. The available values for this parameter as well as the procedure required to change it are described in "Configuring system max values" on page 106.
- Only one dual-tag endpoint can exist on the same port per instance.
- The inner VLAN of the dual-tag endpoint cannot be configured dynamically. In other words, an existing single-tag VLL endpoint cannot be changed to a dual tag VLL endpoint on the fly. You must delete the single-tag VLL endpoint before configuring a dual-tag endpoint.
- A dual tag VLL endpoint neither recognizes nor forwards packets that have a single tag. However, a single-tag endpoint can recognize and forward dual tag packets because the endpoint treats the second tag as data.
- If only the outer VLAN is specified for a given endpoint, the VLAN is called a less-specific VLAN. Similarly, if both the outer and inner VLANs are specified, the VLAN is called a more-specific VLAN (in relation to the outer VLAN).
- If a less-specific VLAN is already configured on a given port, then a more-specific VLAN with the same outer VLAN tag can be configured on that port. In the following example, a less-specific, tagged endpoint has been configured with VLAN 100 on port e 2/1, and a more-specific endpoint with outer-VLAN value of "100" and an inner-VLAN value of "200" is configured on port e 2/1.

  ```
  NetIron(config-mpls)#vll-local test1
  NetIron(config-mpls-vll-lo-test1)#vlan 100
  NetIron(config-mpls-vll-lo-test1-vlan)#tag e 2/1
  NetIron(config-mpls-vll-lo-test1-if-e-2/1)#vlan 100 inner-vlan 200
  NetIron(config-mpls-vll-lo-test1-vlan)#tag e 2/1
  NetIron(config-mpls-vll-lo-test1-vlan)#
  ```

  The result of this example is that single-tagged packets received on port 2/1 with VLAN ID value of "100" and double-tagged packets with an outer-VLAN value of "100" and inner-VLAN of any value other than "200" are sent back out from port 2/1 with outer-VLAN value of "100" and an inner-VLAN value of "200". Dual-tagged packets received on port 2/1 with an outer-VLAN value of "100" and an inner-VLAN value of "200" are sent back out from port 2/1 as single-tagged packets with a VLAN value of "100".

- In any given Local VLL instance, two dual-tag endpoints on the same port are not allowed. The Error messages displayed in **bold** in the following two configuration examples describe this restriction.

```
NetIron(config-mpls)#vll-local test3
NetIron(config-mpls-vll-lo-test3)#vlan 100 inner-vlan 400
NetIron(config-mpls-vll-lo-test3-vlan)#tag e 2/3
NetIron(config-mpls-vll-lo-test3-if-e-2/3)#vlan 100 inner-vlan 500
NetIron(config-mpls-vll-lo-test3-vlan)#tag e 2/3
Error - VLL test3 already has a dual tag end-point on port 2/3 - another dual
tag endpoint on the same port not allowed.
NetIron(config-mpls)#vll-local test4
NetIron(config-mpls-vll-lo-test3)#vlan 1000 inner-vlan 400
NetIron(config-mpls-vll-lo-test3-vlan)#tag e 2/3
NetIron(config-mpls-vll-lo-test3-if-e-2/3)#vlan 2000 inner-vlan 500
NetIron(config-mpls-vll-lo-test3-vlan)#tag e 2/3
Error - VLL test4 already has a dual tag end-point on port 2/3 - another dual
tag endpoint on the same port not allowed.
```

To support dual tags, the VLAN CLI command in the Local VLL configuration mode has a parameter that lets you configure dual tag endpoints.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test1
NetIron(config-mpls-vll-lo-test1)# vlan 200 inner-vlan 300
NetIron(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/2
```

Syntax: [no] vlan <*VLAN-ID*> inner-vlan <*Inner-VLAN-ID*>

The range for **VLAN-ID** is 1 – 4094. (This parameter range excludes the default VLAN ID.)

The range for **inner-VLAN-ID** is 1 – 4095. (This parameter range does not exclude the default VLAN ID.)

Configuring Local VLL QoS (optional)

You can configure a Class of Service (COS) value for either a tagged or untagged port. If the COS value is configured, it is used to determine traffic priority as described in "Local VLL QoS" on page 1482.

To set a COS value for an untagged port use the following command.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test1
NetIron(config-mpls-vll-test1)# untagged ethernet 1/1
NetIron(config-mpls-if-e1000-1/1)# cos 3
```

To set a COS value for an tagged port use the following command.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vll-local test1
NetIron(config-mpls-vll-test1)# vlan 200
NetIron(config-mpls-vll-vlan)# tagged 1/2
NetIron(config-mpls-if-e1000-1/2)# cos 4
```

Syntax: [no] cos <*cos-value*>

The <*cos-value*> can be set to a priority between 0 - 7.

Displaying Local VLL information

You can display the following information about the Local VLL configuration on a NetIron router:

- Information about individual Local VLLs configured on the router

- Information about VLL Endpoint Statistics

Displaying information about Local VLLs

To display brief information about Local VLLs.

```
NetIron# show mpls vll-local
Name         VLL-ID    End-point1                      End-point2          State
foundrylong   1        tag vlan 100  e 5/12            undefined           DOWN
vlllocalfou
ndrylongvll
localfoundr
ylongvlloc
alfoundry

test          2      tag vlan 200 inner-vlan 50 e 2/1  tag vlan 200 e 2/2  UP
NetIron
```

Syntax: show mpls vll-local

For each Local VLL on the device, the following information is displayed.

TABLE 249 Output from the show mpls vll-local brief command

This field...	Displays...
Name	The configured name of the Local VLL.
VLL-ID	The VLL ID.
End-point	How packets are forwarded out of the egress port of the Local VLL. This can be one of the following: • "untagged <*portnum*>" – Forward the packet out the specified port as untagged. • "tag VLAN <*vlan_id*> <*portnum*>" – Tag the packet with the specified VLAN ID and forward the packet out the specified port. • "undefined" – An endpoint has not been configured for this Local VLL • "inner-vlan" – Describes the inner-VLAN tag for an end-point that is configured for dual-tagging.
State	The current state of the Local VLL. This can be either UP or DOWN. Data can be forwarded over the Local VLL only when the state is UP.

To display detailed information about a specific Local VLL configured on the device.

```
NetIron# show mpls vll-local detail
VLL test-1 VLL-ID 1 IFL-ID --
  State: UP
  End-point 1:      untagged e 2/2
                    COS: --
  End-point 2:      untagged e 2/13
                    COS: --
VLL test-2 VLL-ID 2 IFL-ID --
  State: UP
  End-point 1:      tagged  vlan 2500  e 2/10
                    COS: --
  End-point 2:      tagged  vlan 2500  e 2/9
                    COS: --
VLL test-3 VLL-ID 3 IFL-ID --
  State: UP
  End-point 1:      tagged  vlan 2501  e 2/10
                    COS: 6
  End-point 2:      tagged  vlan 2501  e 2/9
                    COS: 5
Vll test-4 VLL-ID 4 IFL-ID 4096
  State: UP
    End-point 1:    tagged  vlan 100   inner-vlan 45 e 2/1
                    COS: --
    End-point 2:    tagged  vlan 100  e 2/3
                    COS: --
```

Syntax: show mpls vll-local detail | *<vll-name>*

The detail parameter displays detailed information for all Local VLLs in the router while specifying a particular VLL using the *<vll-name>* option limits the display to the specified Local VLL.

For each configured Local VLL, the command displays the following information in addition to the information described in Table 249.

TABLE 250 Output from the show mpls vll-local detail command

This field...	Displays...
IFL-ID	The Internal Forwarding Lookup Indentifier that is allocated to each Local VLL instance that has at least one dual tag endpoint. For instances that do not have dual tag endpoints, the IFL-ID is displayed as "--".
state	The current state of the VLL. This can be one of the following: • "UP": Local VLL is operational - packets can flow. • "DOWN - configuration incomplete": A required configuration statement is missing. • "DOWN - endpoint port is down": The physical endpoint port is down due to a link outage or is administratively disabled.

TABLE 250 Output from the show mpls vll-local detail command (Continued)

This field...	Displays...
End-point	How packets are forwarded out of the egress port of the Local VLL. This can be one of the following: • "u"ntagged <*portnum*>" – Forward the packet out the specified port as untagged. • "tag VLAN <*vlan_id*> <*portnum*>" – Tag the packet with the specified VLAN ID and forward the packet out the specified port. • "undefined" – An endpoint has not been configured for this Local VLL • "inner-vlan" – Describes the inner-VLAN tag for an end-point that is configured for dual-tagging.
COS	The optional COS setting for the Local VLL. If a COS value is set, the COS value can be between 0 - 7.

Displaying Local VLL endpoint statistics

To view the forwarding counters for each Local VLL configured on the system, you can display Local VLL Endpoint traffic statistics. The display is shown such that for a given port range that receives traffic, how many packets are arriving from the customer endpoint.

NOTE
When the forwarding cam is full, the vll-local software forwarded packets are not accounted in vll-local statistics.

To display all VLL traffic statistics on a NetIron router, enter the following command.

```
NetIron# show mpls stat vll-local
VLL-Name       End-point 1/2       VLL Port(s)      VLL-Ingress-Pkts
-------------- ----------------  --------------    ---------------
test             End-point1          e2/3-2/4         835192705
                 End-point2          e2/3-2/4         838181595
test1            End-point1          e2/3-2/4         544017
                 End-point2          e2/3-2/4         544017
test3            End-point1          e2/1             544022
                 End-point2          e2/2             544022
```

To display VLL traffic statistics for a VLL instance specified by its VLL name, enter the following command.

```
NetIron# show mpls stat vll-local test
VLL-Name       End-point 1/2       VLL Port(s)         VLL-Ingress-Pkts
---------       ----------------     ---------------       -----------------
test           End-point1          e2/3-2/4            0
               End-point2          e2/3-2/4            0
```

To display Local VLL traffic statistics for a VLL instance specified by its VLL ID, enter the following command.

```
NetIron# show mpls stat vll-local 4
VLL-Name       End-point 1/2       VLL Port(s)         VLL-Ingress-Pkts
---------       ---------------       --------------       ------------------
test3          End-point1          e2/1                0
               End-point2          e2/2                0
```

Syntax: show mpls statistics vll-local [<*vll-name*> | <*vll-id*>]

The <*vll-name*> variable is the configured name for a Local VLL instance.
The <*vll-id*> variable is the ID of a VLL instance.

For following information is displayed in the **show mpls statistics vll** command:

TABLE 251 Output from the **show mpls vll-local command**

This field...	Displays...
VLL-Name	The configured name of the Local VLL instance.
End-point 1/2	Either the End-point1 or End-point2 of the Local VLL instance
VLL Ports	The port or ports that are assigned to the end point. If there are multiple ports, they are members of a trunk.
VLL-Ingress-Pkts	Packets arriving on the specified end point from outside the Local VLL.

Clearing the VLL traffic statistics

To clear the entries stored for all Local VLL statistics, enter a command such as the following.

```
NetIron# clear mpls statistics vll-local
```

Syntax: clear mpls statistics vll-local

Enabling MPLS Local VLL traps

You can enable trap notification to be sent for Local VLLs by entering the following command.

```
NetIron(config)# snmp-server enable trap mpls vll-local
```

Syntax: [no] snmp-server enable trap mpls vll-local

Refer to the *IronWarre MIB Reference* for MPLS VLL trap notifications.

Disabling Syslog messages for MPLS VLL-local and VLL

Transitions of VLL local instances from an up state to a down state and vice versa are logged by default. You can disable the logging of these events by entering the following command.

```
NetIron(config)# no logging enable mpls
```

Syntax: [no] logging enable mpls

Similarly, the generation of Syslog message for MPLS VLL events are enabled by default. You can disable the logging of these event by entering the following command.

```
NetIron(config)# no logging enable mpls vll
```

Syntax: [no] logging enable mpls vll

Refer to Table 454 on page 2299 for the Syslog messages generated.

Configuring MPLS Virtual Private LAN Services 33

Overview

This chapter explains how to configure Virtual Private LAN Services (VPLS). VPLS is a method for carrying Layer 2 frames between customer edge (CE) devices across an MPLS domain. The implementation supports VPLS as described in the IETF RFC 4762 (Virtual Private LAN Services over MPLS Using LDP Signaling).

NOTE
VPLS should not be used if Foundry Discovery Protocol (FDP) is configured.

The following list displays the Virtual Private LAN Services (VPLS) features supported by PowerConnect B-MLXe:

- Virtual Private LAN Services (VPLS)
- Per-VPLS MAC Table Limit
- Maximum Number of MAC Entries for a VPLS instance
- LSP to Reach a Peer within a VPLS instance
- LSP Load Balancing for VPLS Traffic
- Dual tag support for VPLS and Local VPLS
- VPLS Broadcast/Multicast/Unknown-Unicast Packet Limiting
- Flooding Layer2 BPDUs in VPLS
- VPLS Tagged mode
- VPLS CPU Protection
- VLAN Translation
- VPLS MTUs
- Dynamic LAG support for VPLS endpoints
- Flooding Layer 2 BPDUs with a VPLS Instance
- VPLS MTU Enforcement
- VPLS Local Switching
- MPLS VPLS Traps
- Disabling Syslog Messages for MPLS VPLS
- Local VPLS
- VC label allocation managed by MPLS
- VPLS LDP
- VPLS FID sharing

How VPLS works

Virtual Private LAN Services (VPLS) enhances the point-to-point connectivity defined in the Draft-Martini IETF documents by specifying a method for virtual circuits (VCs) to provide point-to-multipoint connectivity across the MPLS domain, allowing traffic to flow between remotely connected sites as if the sites were connected by a Layer 2 switch.

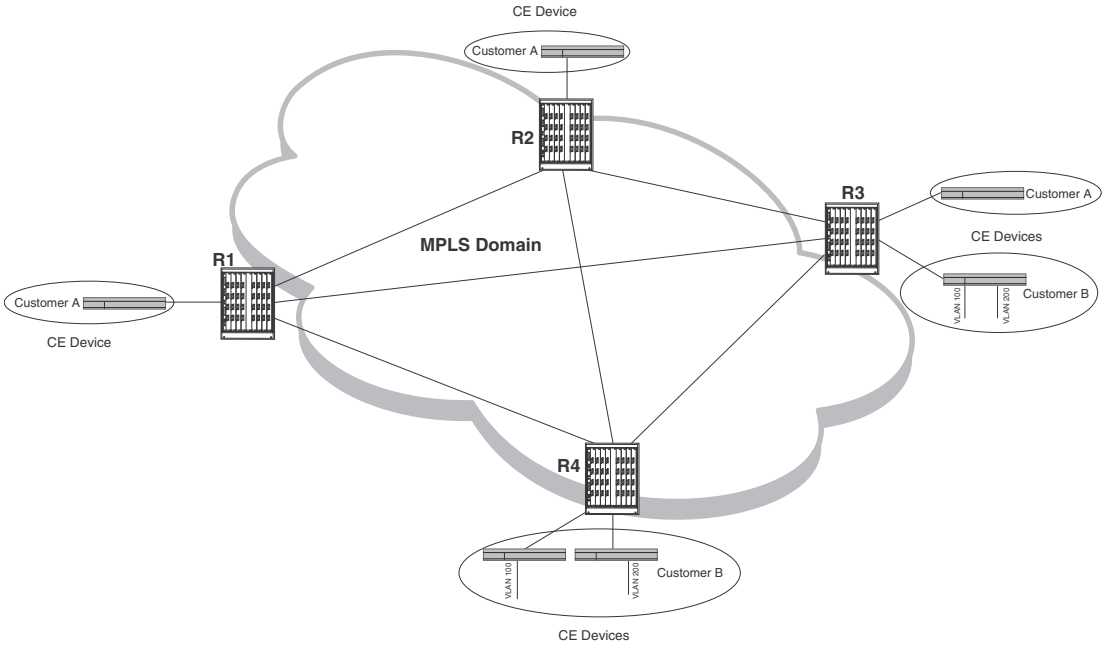
VPLS can be used to transport Ethernet frames to and from multiple, geographically dispersed sites belonging to a customer Virtual Private Network (VPN). The Provider Edge (PE) devices connecting the customer sites provide functions similar to a Layer 2 switch. The PE devices learn the MAC addresses of locally connected customer devices, flood broadcast and unknown unicast frames to other PE devices in the VPN, and create associations between remote MAC addresses and the VC Label Switch Patchs (LSPs) used to reach them.

Figure 198 shows an illustration of a VPLS configuration with two customer VPNs. Two separate VPLS instances have been created, one for Customer A's VPN and one for Customer B's VPN. A VPLS instance consists of a full mesh of VC LSPs between the customers' PE devices. In the example, Customer A's VPLS instance consists of VC LSPs between routers R1, R2, and R3. Customer B's VPLS instance consists of VC LSPs between routers R3 and R4. Because VC LSPs are unidirectional, separate VC LSPs exist in each direction between each of the PE devices. When Label Distribution Protocol (LDP) is enabled on the MPLS interfaces on the PE devices, the VC LSPs are established automatically through LDP when you specify the VPLS peers on the PE devices.

Alternatively, LSPs can be established using Resource ReSerVation Protocol- Traffic Engineering (RSVP-TE) by manually configuring LSPs to all PE devices. The same LSP from one PE to another PE can be shared by multiple VPLS instances for traffic belonging to different customers. In this case, traffic belonging to different customers has the same tunnel label, but different VC labels. If more than one LSP exists from one PE to another PE for multiple VPLS instances, traffic belonging to the different VPLS instances can be load-balanced across the LSPs. In this case, traffic belonging to the different VPLS instances has different tunnel and VC labels.

In Figure 198, the VPLS instance for Customer A links its CE devices so that they appear to be a single Layer 2 broadcast domain. The VPLS instance for Customer B has two VLANs configured within the VPLS instance, VLAN 100 and VLAN 200. The VPLS instance for Customer B has two endpoints on PE device R4. Unlike a Virtual Leased Line (VLL), a VPLS instance can have multiple endpoints. The PE device performs local and remote VLAN tag translation, so that multiple VLANs can be specified under a single VPLS instance.

FIGURE 198 Sample VPLS configuration



A PE device in the VPLS configuration operates like a standard Layer 2 switch, in that it performs MAC address learning, flooding, and forwarding for the CE devices in each VPLS instance. For example, when PE device R1 receives a Layer 2 frame with a given MAC destination address from Customer A's CE device, it looks up the MAC address in a Layer 2 forwarding table that records associations between MAC addresses and VC LSPs. This forwarding table is known as the *VPLS MAC database.*

If the MAC address is found in the VPLS MAC database, the PE device finds the associated VC LSP, encapsulates the frame as an MPLS packet, and pushes an inner VC label and outer tunnel label onto the packet. The packet is then sent over a tunnel LSP to the VC peer. If the MAC address is not found in the VPLS MAC database, the frame is flooded to all of the PE devices and locally connected CE devices (except for the CE device that originated the frame) in the customer's VPLS instance. When a response is received, an entry for the MAC address and the VC from which it arrived is added to the VPLS MAC database. Subsequent frames targeting the MAC address are not flooded to the other devices in the VPLS instance. In this way, the PE device learns the MAC addresses of the remotely connected customer devices. MAC addresses received at the local VPLS endpoints are also learned in the VPLS MAC database for the VPLS instance.

The PE devices do not run Spanning Tree Protocol (STP) over the MPLS domain. The full mesh of PE devices in a VPLS configuration allows one PE device to reach any other PE device in the VPN in exactly one hop, with no transit PE devices in between. The PE devices apply a split horizon rule when forwarding frames within the VPN. When a PE receives a customer frame from a VC LSP, it can forward the frame only to a directly attached customer device, not to another VC LSP. This allows the VPLS instance to have a loop-free topology without having to run STP.

NOTE
Packets that arrive on an interface with the same destination MAC address as the interface are forwarded in hardware just like packets with other destination addresses.

Configuring VPLS instances

This section explains how to set up VPLS instances.

Creating a VPLS instance

You create a VPLS instance by entering VPLS configuration statements on two or more PE routers. The endpoints of a VPLS instance are associated by having the same VPLS Virtual Circuit Identifier (VCID) on each PE router.

To create a VPLS instance, enter commands such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls v1 100
NetIron(config-mpls-vpls-v1)#
```

On the VPLS peers (if they are devices), you would enter commands such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls v1 100
NetIron(config-mpls-vpls-v1)#
```

Syntax: **vpls** <*name*> <*vpls-vcid*> [**cos** <*cos-value*>] [**max-mac** <*max-mac-entries*>]

The **vpls** <*name*> variable specifies the VPLS instance name.

The <*vpls-vcid*> variable is the VPLS ID number of the VPLS instance. The <*vpls-vcid*> variable can take a value in the range of 1 through 4294967294.

You can optionally specify a Class of Service (CoS) setting for the VPLS instance. If a CoS value is set, the device selects a tunnel LSP that also has the CoS value if one is available. If no tunnel LSP with this CoS value is available, the device selects a tunnel LSP with the highest configured CoS value (although never higher than the CoS setting for the VPLS instance). The CoS value has a range from 0 through 7.

Setting a per-VPLS MAC table limit

In previous versions of the IronWare Multi-Service software, you could set the maximum size of the VPLS MAC database and set a soft limit on the amount of memory resources available for each VPLS from the VPLS MAC database. While this soft limit helped optimize the amount of resources for each VPLS instance, it did not restrict an individual instance of a VPLS from learning more entries than the amount set for the specified VPLS instance. In practice, it would allow an instance to grow its individual entries up to the amount available in the VPLS MAC database. Under these circumstances, an individual VPLS instance could consume all of the resources and starve other VPLS instances.

You can configure a maximum number of MAC entries that can be learned for a specified VPLS instance. This number cannot be exceeded. This limit can be configured at any time, although operation is more robust if you configure the limit at the same time that you configure the VPLS instance.

Configuring the maximum number of MAC entries for a VPLS

To configure a maximum number of MAC entries available to a VPLS instance, enter commands such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls v1 100 max-mac 3000
```

Syntax: vpls <*name*> <*vpls-vcid*> **[cos** <*cos-value*>**] [max-mac** <*max-mac-entries*>**]**

The <*name*> variable is the name of the VPLS instance for which you are configuring the maximum number of MAC entries.

The <*vpls-vcid*> variable is the VPLS ID number of the VPLS instance for which you are configuring the maximum number of MAC entries. The <*vpls-vcid*> variable can take a value in the range of 1 through 4294967294.

You can optionally specify a Class of Service (CoS) setting for the VPLS instance. If a CoS value is set, the device selects a tunnel LSP that also has this CoS value, if one is available. If no tunnel LSP with this CoS value is available, the device selects a tunnel LSP with the highest configured CoS value (although never higher than the CoS setting for the VPLS instance). The CoS value has a range from 0 through 7.

The <*max-mac-entries*> variable specifies the maximum number of MAC entries that can be learned for the VPLS instance. The <*max-mac-entries*> value can range from 1 to global VPLS MAC database size.

Specifying the maximum size of the VPLS MAC database

The VPLS MAC database serves as a Layer 2 forwarding table that associates local MAC addresses with CE devices and remote MAC addresses with VC LSPs used to reach the remote CE devices. The minimum, maximum, and default values for this parameter are described in Table 15. This number represents the total number of MAC addresses that can be learned for all VPLS instances configured on the device.

You can globally specify a different maximum size for the VPLS MAC database by entering a command such as the following.

```
NetIron(config)# system-max vpls-mac 4096
```

Syntax: system-max vpls-mac <*number-of-entries*>

NOTE
You must reload your system for the **system-max vpls-mac** command to take effect.

Clearing the contents of the VPLS MAC database

To clear the entries stored in the VPLS MAC database belonging to a VPLS instance, enter a command such as the following.

```
NetIron# clear mac vpls name v1
```

Syntax: clear mac vpls name <*name*> **| id** <*vpls-vcid*> **| ethernet** <*portnum*> **| label** <*label*>

The **name** <*name*> parameter clears all entries associated with the named VPLS instance.

The **id** <*vpls-vcid*> parameter clears all entries associated with the specified VPLS VCID.

The **ethernet** <*portnum*> parameter clears all local MAC entries on the specified port.

The **label** <*label*> parameter clears all entries associated with a local VC label.

Specifying the maximum number of VPLS instances on the device

The minimum, maximum and default values for this parameter are described in Table 15. The configured maximum number of VPLS instances has an effect on the size of the label range for each VPLS instance. The label range is the set of labels that the VPLS instance can assign to its peers for use as the peer's local VC label.

The product of the maximum number of VPLS instances and the label range will always equal 65536. This means that if the maximum number of VPLS instances is 2048, then the label range is 32; if the maximum number of VPLS instances is 8192, then the label range is 8; and so on.

To change the maximum number of number of VPLS instances to 8192, enter the following command.

```
NetIron(config)# system-max vpls-num 8192
```

Syntax: system-max vpls-num <*number-of-VPLS-instances*>

NOTE
You must reload your system for this command to take effect.

You can display the configured maximum number of VPLS instances, as well as the size of the label range, with the **show mpls vpls summary** command.

Specifying VPLS peers

NOTE
Starting with release 04.0.00, the implementation of *BGP-based auto-discovery for VPLS* (also called *VPLS auto-discovery*) eliminates the need for manual configuration of VPLS peers for every VPLS instance configured on the device. For details, refer to chapter 34, "Configuring BGP-Based Auto-Discovery for VPLS".

As part of the VPLS configuration, you specify the IP address of each VPLS peer. VPLS requires a full mesh of tunnel LSPs; each PE router must have tunnel LSP reachability to each of its VPLS peers. Tunnel LSP reachability is defined as having at least one operational RSVP- or LDP-signalled LSP with the destination (the 'to" address of the LSP) matching the VPLS peer's IP address. An LSP terminating on the VPLS peer but configured with a different destination address would not be considered a match.

By default, each PE router attempts to initiate an LDP session through extended discovery with its VPLS peers, if a session is not already established. Each VPLS instance is allocated a range of 32 labels. The PE router assigns one label in the range to each of its peers to be used as the peer's local VC label. If there are more than 32 peers in the VPLS instance, an additional label range is automatically allocated to the VPLS instance. The size of the label range depends on the configured maximum number of VPLS instances on the device. Refer to "Specifying the maximum number of VPLS instances on the device" on page 1498 for more information.

Once the LDP session is established, the PE device advertises the local VC label, along with the VPLS ID, to its VPLS peers in a downstream-unsolicited manner. In a similar way, the PE also learns the remotely assigned VC labels from its VPLS peers.

To specify three remote VPLS peers within a VPLS instance, enter a command such as the following.

```
NetIron(config-mpls-vpls-v1)# vpls-peer 192.168.2.100 192.168.2.101 192.168.2.102
```

Syntax: vpls-peer <*ip-addr*> [<*ip-addr*>...]

The IP address of each VPLS peer must match that of a destination for a tunnel LSP configured on the device.

Setting the VPLS VC mode

The PowerConnect B-MLXe routers support the following VPLS VC modes, which determine whether or not VLAN tags are carried across the MPLS cloud:

- Raw mode – This is the default VC mode. When this mode is in effect, the VLAN tag information in the original payload *is not* carried across the MPLS cloud.
- Tagged mode – When tagged mode is enabled, the VLAN tag information in the original payload is carried across the MPLS cloud.

VPLS raw mode

By default, VPLS packets are sent to remote peers over the MPLS cloud in *raw mode*. This means that no VLAN tag information in the payload is carried across the MPLS cloud. In raw mode, the VLAN priority (Class of Service) of the original (incoming) packets is lost once the packets are sent through the cloud.

NOTE
If desired, you can enable the device to preserve the VLAN tag information in the payload and carry it across the MPLS cloud to remote peers. For more information, see "VPLS tagged mode" on page 1500.

CoS behavior for VPLS raw mode

NOTE
This section assumes that you understand how QoS works. For details, see "Configuring Quality of Service for the NetIron MLX" on page 283.

Table 252 describes the expected Class of Service (CoS) behavior for VPLS packets when VPLS raw mode is in effect.

TABLE 252 Expected class of service behavior for VPLS raw mode

VPLS endpoints	Incoming packet		MPLS cloud		Outgoing packet	
	Outer VLAN	Inner VLAN	Tunnel/VC label (Z)	Payload tag	Outer VLAN	Inner VLAN
Dual-tagged to dual-tagged	X	Y	V or internal priority	N/A	W or Z	Z
Single-tagged to dual-tagged	X	N/A				Z
Untagged to dual-tagged	N/A	N/A				Z
Dual-tagged to single-tagged	X	Y				N/A

Legend for Table 252

V = Mapped EXP bits from internal priority (X contributes to internal priority) using the EXP encode table. Table 57 shows the default EXP encode table.

W = Mapped CoS from internal priority (Z contributes to internal priority) using the CoS encode table.

X = Original outer VLAN CoS.

Y = Original inner VLAN CoS.

Z = Incoming EXP bits as described by the *Tunnel / VC label* column = V or internal priority.

- The *Tunnel/VC label* column differentiates the behavior when **qos exp encode** policy is ON (default) or OFF.
- The *Outgoing packet Outer VLAN* column differentiates the behavior when **qos pcp encode** policy is ON (default) or OFF.

VPLS tagged mode

VPLS tagged mode enables the preservation of the VLAN tag information in the payload. In VPLS tagged mode, the VLAN priority of the original (incoming) packets is carried across the MPLS cloud to remote peers.

By default, VPLS packets are sent across the MPLS cloud in raw mode. To use VPLS tagged mode, enable it per VPLS instance on both sides of the communicating edge routers. When this feature is enabled, the VLAN tag is determined as follows:

- If the original packet has one VLAN tag, the payload tag will be the (outer) VLAN tag of the original packet.
- If the original packet has dual VLAN tags, the payload tag will be the inner VLAN tag of the original packet.
- If the original packet is untagged, the payload tag will be the configured VLAN on the VPLS untagged endpoint, and the CoS will be 0.
- If the original packet has an I-component Service Identifier (ISID) tag, the payload tag will be the unmodified ISID tag.

For more information about CoS behavior for VPLS tagged mode, see Table 253 on page 1501.

VPLS tagged mode must be enabled on both sides of the communicating edge routers. If the VPLS VC type does not match, the remote peer will not transition into operational state. Because each remote peer has its own operational state, the impact may differ from one remote peer to another, depending on its current state. The remote peer state can be categorized into two general categories, as follows:

- *Remote peer in operational state* – If the remote peer is in operational state, a VC withdraw message and a new VC bind message will be sent to the remote peer to tear down the current VC binding and to communicate the new VC type, respectively. This scenario assumes that the remote router is also a Dell router running the same code level. The VC tear-down and re-bind should cause the remote peer to transition its peer state to “VC Parameter Check” state, because its own VC type will now be mismatched with that of the new VC type received. Once the same tagged mode configuration is also applied to the remote router, the peer state for both routers should transition into operational state. As part of the VC tear-down, the hardware forwarding entries on the Interface module (LP) will be cleaned up. When the peer transitions to operational state, its hardware forwarding entries will be reprogrammed based on its tagged mode setting.

- *Remote peer not in operational state* – The category indicates that the VC has not yet been formed with the VPLS peer on the remote router. Remote peers may be in this category for many reasons (for example, “No local port defined”, “No Tunnel”, “No LDP Session”, “VC Parameter Check”, and so on). In this scenario, there is no need to tear down the VC binding. When the VPLS tagged mode configuration changes, most of the peers in this category will not change their operational state or perform any actions triggered by this configuration change. For remote peers that are in the state “VC Parameter Check” state because of a VC type mismatch, the configuration change will trigger the sending of a VC bind message with the new VC type to the remote router. If the remote peer’s VC type becomes compatible due to this configuration change and there is no other VC parameter mismatch, then the state of the remote peer will transition to operational state.

To enable VPLS tagged mode, see “Configuring VPLS tagged mode” on page 1511.

CoS behavior for VPLS tagged mode

NOTE
This section assumes that you understand how QoS works. For details, see “Configuring Quality of Service for the NetIron MLX” on page 283.

Table 253 describes the expected Class of Service (CoS) behavior for VPLS packets when VPLS tagged mode is enabled.

TABLE 253 Expected class of service behavior for VPLS tagged mode

VPLS endpoints	Incoming packet		MPLS cloud		Outgoing packet	
	Outer VLAN	Inner VLAN	Tunnel/VC label (Z)	Payload tag	Outer VLAN	Inner VLAN
Dual-tagged to dual-tagged	X	Y	V or internal priority	Y	W or Y	Y
Single-tagged to dual-tagged	X	N/A		X	W or X	X
Untagged to dual-tagged	N/A	N/A		0	W or 0	0
Dual-tagged to single-tagged	X	Y		Y	W or Y	N/A

Legend for Table 253

V = Mapped EXP bits from internal priority (X contributes to internal priority) using the EXP encode table. Table 58 shows the default EXP encode table.

W = Mapped CoS from internal priority (Z contributes to internal priority) using the CoS encode table.

X = Original outer VLAN CoS.

Y = Original inner VLAN CoS.

Z = Incoming EXP bits as described by the *Tunnel / VC label* column = V or internal priority.

- The *Tunnel/VC label* column differentiates the behavior between when **qos exp encode** policy is ON (default) or OFF.
- The *Outgoing packet Outer VLAN* column differentiates the behavior between when **qos pcp encode** policy is ON (default) or OFF.

QoS for VPLS traffic

By default, packets travelling through an MPLS domain are treated equally from a QoS standpoint, in a best effort manner. However, if a Layer 2 packet has an internal priority in its 802.1q tag, or the LSP or VPLS to which the packet is assigned has a configured Class of Service (COS) value, QoS can be applied to the packet in the MPLS domain. The internal priority or COS value is mapped to a value in the EXP field of the packet's MPLS header. The value in the EXP field is then mapped to an internal forwarding priority, and the packet is sent to the hardware forwarding queue that corresponds to the internal forwarding priority.

QoS for VPLS traffic at the ingress LER

The following methods can be used to provide QoS to packets entering a VPLS:

- Use the COS value assigned to the tunnel LSP used to reach the VPLS peer.

  When a tunnel LSP has a user-configured COS value, all packets in all VPLS travelling through the tunnel LSP receive the same QoS.

- Use the COS value assigned to the VPLS.

  If a COS value is set for the VPLS, the device selects a tunnel LSP that also has this COS value, if one is available. If no tunnel LSP with this COS value is available, the device selects a tunnel LSP with the highest configured COS value (although never higher than the COS setting for the VPLS).

  If the selected tunnel LSP does not have a COS value, the VPLS configured COS value is used to provide QoS. The VPLS COS value is mapped to a value in the EXP field. This allows traffic multiple VPLS using a single tunnel LSP, traffic from each VPLS can receive different QoS treatment.

- Use the priority in the packet's 802.1q tag.

  When neither the tunnel LSP nor the VPLS has a configured COS value, the device examines the priority in the Layer 2 packet's 802.1q tag, if the packet has one. Consequently, Layer 2 packets with the same 802.1q priority receive the same QoS in the VPLS.

- Use the configured priority of the port.

  If neither the tunnel LSP nor the VPLS has a configured COS value, and the Layer 2 packet does not have an 802.1q priority, QoS can be provided based on the priority of the incoming port. A port can be assigned a priority from 0 (lowest priority) to 7 (highest priority). The default port priority is 0.

  By assigning different priorities to the ports where customer edge (CE) devices are connected (that is, the VPLS endpoints), you can provide QoS to untagged Layer 2 traffic received from different customer locations.

When a packet enters a VPLS, the PE router that serves as both the VPLS endpoint and the ingress of a tunnel LSP pushes two labels onto the packet the inner VC label and the outer tunnel label. The packet's priority resides in the EXP field of the MPLS label header. The VC label and the tunnel label carry the same value in the EXP field.

The following table lists how a Layer 2 packet's priority is mapped to a value in the EXP field and how the EXP value is mapped to a priority queue.

Tunnel LSP configured COS or VPLS configured COS or 802.1q priority or Configured port priority	Value placed in the tunnel and VC label EXP field	Priority queue
7	7	qosp7 (highest priority)
6	6	qosp6
5	5	qosp5
4	4	qosp4
3	3	qosp3
2	2	qosp2
1	1	qosp1
0	0	qosp0 (best effort)

Specifying an LSP to reach a peer within a VPLS

You can specify the LSPs that can be used to reach a peer within a VPLS. You can specify up to four Resource ReSerVation Protocol (RSVP) LSPs per VPLS peer. VPLS subsequently selects one of the LSPs configured to reach the specified peer. Any of the configured LSPs can be used, and the order of configuration is not relevant to the selection of the LSP. If none of the assigned LSPs is operational, the VPLS session with the peer is down. An LSP is considered down when the LSP's primary, secondary, and detour paths are all down.

RSVP LSPs must be pre-configured prior to their assignment to the VPLS peer. Additionally, the VPLS peer's IP address must match the target IP address of any RSVP LSPs assigned to it. If these addresses do not match, the configuration will be rejected. An LSP that is assigned to any VPLS will not be allowed to be deleted from the configuration unless the VPLS LSP assignment is deleted first. If no LSPs have been assigned to a VPLS peer, the existing mechanism is used to select an appropriate LSP for the peer.

When LSP assignment is configured, ignore the configured COS of the LSP, and ignore the VPLS to select an LSP for the VPLS peer. However, traffic sent on the LSP will use the CoS of the LSP. If LSP load balancing is enabled for a VPLS peer, traffic is load-balanced on all assigned LSPs that are operational.

To specify LSPs for a VPLS peer within a VPLS instance, enter a command such as the following.

```
NetIron(config-mpls-vpls-v1)# vpls-peer 192.168.2.100 lsp t1 t2 t3 t4
```

Syntax: vpls-peer <*ip-address*> **lsp** <*lsp1*> [<*lsp2*> <*lsp3*> <*lsp4*>]

The <*ip-address*> variable specifies the IP address of the VPLS peer to which you want to assign LSPs.

The <*lsp1*> <*lsp2*> <*lsp3*> <*lsp4*> variables are the names of the LSPs that you want to assign to the VPLS peer. You can assign up to four LSPs to a peer using this command. If a VPLS peer is not assigned any LSPs, the default mechanisms for selecting an LSP for the VPLS peer are used.

LSP load balancing for VPLS traffic

In a VPLS instance, traffic from one VPLS peer to another is forwarded over an MPLS tunnel LSP. If more than one tunnel LSP exists from the device to a VPLS peer, the device can select multiple tunnel LSPs to forward VPLS traffic to the peer. Known unicast traffic is load-balanced across the selected tunnel LSPs. Broadcast and unknown unicast traffic is always sent over a single tunnel LSP, however.

For VPLS LSP load-balancing, select an LSP based on a hash-index which is calculated as follows:

NOTE
For VPLS traffic, source and destination MAC addresses come from the inner customer Ethernet header.

- **Layer-2, non-IPv4, and IPv6 packets:** Source MAC address and destination MAC address.
- **IPv4, non-TCP/UDP packets**: Source MAC address and destination MAC address, source IP address and destination IP address.
- **IPv4 TCP packets:** Source MAC address and destination MAC address, source IP address and destination IP address, and TCP source port and TCP destination port.
- **IPv4 UDP packets:** Source MAC address and destination MAC address, source IP address and destination IP address, and UDP source port and UDP destination port.
- **IPv6 non-TCP/UDP packets**: Source MAC address and destination MAC address, source IP address and destination IP address.
- **IPv6 TCP packets:** Source MAC address and destination MAC address, source IP address and destination IP address, and TCP source port and TCP destination port.
- **IPv6 UDP packets:** Source MAC address and destination MAC address, source IP address and destination IP address, and UDP source port and UDP destination port.

A tunnel LSP's eligibility for load balancing depends on whether CoS values are defined for the VPLS instance and the tunnel LSP:

- If the VPLS instance does not have a CoS value defined, then all tunnel LSPs to the peer are eligible for load balancing.
- If a VPLS instance has a CoS value defined, and at least one tunnel LSP to the peer has a CoS value less than or equal to the VPLS instance CoS value, then all tunnel LSPs with the highest CoS value less than or equal to the VPLS instance CoS value are eligible for load balancing.
- If a VPLS instance has a CoS value defined, and none of the tunnel LSPs to the peer has a CoS value less than or equal to the VPLS instance CoS value, then all tunnel LSPs to the peer that do not have a CoS value are eligible for load balancing.

NOTE
The LSP's picked for load-balancing must have the same COS values. For example: If COS of LSP1 = 4, LSP2 = 4, LSP3 = 2, LSP4 =2, LSP5 = 1 and VPLS instance COS = 3. Then traffic is load balanced with LSP3 and LSP4 which has same COS values.

LSP load balancing

The device evenly distributes VPLS traffic across tunnel LSPs.

In early software releases, VPLS traffic was unevenly balanced across tunnel LSPs if exactly three tunnels were used for load balancing. For example, for tunnels A, B, and C, VPLS traffic might be distributed among the tunnels as follows: A: 50%, B: 25%, and C: 25%.

Now, the tunnels are fully utilized. Using the same example above, VPLS traffic might be distributed among tunnels A, B, and C as follows: A: 33.3%, B: 33.3%, and C: 33.3%. These percentages are based on a fully distributed hash index generated by the incoming traffic. Actual distribution percentages may vary and are based on the hash index.

Configuring LSP load balancing for VPLS traffic

To configure a VPLS instance to load balance known unicast traffic sent to a VPLS peer across multiple tunnel LSPs, enter a command such as the following.

```
NetIron(config-mpls-vpls-v1) vpls-peer 192.168.9.210 load-balance
```

Syntax: [**no**] **vpls-peer** <*ip-addr*> [**load-balance**]

NOTE
To disable the LSP load balancing, you must delete the VPLS peer with the **no vpls-peer** command, then re-enter the **vpls-peer** command without the **load-balance** option.

NOTE
To disable LSP load balancing when VPLS auto-discovery is enabled on the device, refer to "Disabling load balancing" on page 1550.

In the prior example, when the **load-balance** option is specified, VPLS traffic originating from the device and sent to peer 192.168.9.210 is load balanced across eligible tunnel LSPs whose destination is the peer.

Specifying the endpoint of a VPLS instance

When you configure the VPLS endpoint, you specify what happens to packets exiting the VPLS instance, which VLAN the packet belongs to, as well as whether it is transmitted from the PE device to the CE device over a dual-tagged, single-tagged, or untagged port. You can also specify a server Link Aggregation Group (LAG) group as the endpoint of a VPLS instance.

A VPLS instance can be configured between any combination of dual-tagged, single-tagged, and untagged endpoints. For dual-tagged ports, traffic flows between the dual-tagged endpoint and the MPLS cloud are also supported for traffic switched between a local endpoint and remote peers.

NOTE
Unless VPLS tagged mode is enabled, VPLS will operate in raw mode, meaning no VLAN tags will be carried across the MPLS cloud to remote peers. For more information, see "Configuring VPLS tagged mode" on page 1511.

The Customer Edge (CE) device is connected to the PE router over one or more dual-tagged, single-tagged, or tagged ports.

- With a *single-tagged* port, each pair (port, VLAN ID) is identified as a unique endpoint. If VPLS raw mode is in effect, the tag is of significance between the CE and the PE and is not sent across the MPLS cloud. If VPLS tagged mode is enabled, the tag will be sent across the MPLS cloud.

- In the case of an *untagged* port, an endpoint is identified by the physical port alone, and the packets are sent in untagged Ethernet format within the MPLS payload.
- In the case of a *dual-tagged* port, the packets contain both an outer VLAN tag and an inner VLAN tag. In this configuration, an endpoint can receive packets with two tags and forward them to the other endpoint either single-tagged or dual-tagged. If VPLS tagged mode is enabled, the inner VLAN tag will be sent across the MPLS cloud.

All VPLS endpoints can be dual mode ports (tagged-untagged). An untagged endpoint port is removed from the default VLAN ID 1 and cannot be added back to the default VLAN. A VPLS endpoint can be tagged in multiple VPLS and Layer 2 VLANs and untagged in one other VLAN.

Special considerations for dual-tagged endpoints

Before configuring a dual-tagged VPLS endpoint, consider the following:

- The tag protocol identifier (TPID) of the inner VLAN tag must be 0x8100 be classified as dual-tagged and recognized by dual-tagged endpoints. If the TPID is not 0x8100, the packet will be classified as a single-tagged packet.
- The TPID of the outer VLAN tag must be the port's configured tag type (the default tag type is 0x8100).
- The System Max value for the Internal Forwarding Lookup (IFL) CAM partition must not be set to 0 (zero). If it is set to zero, an informational message such as the following will be displayed.

```
NetIron(config-mpls)#vpls test 10
NetIron(config-mpls-vpls-test10)#vlan 100 inner-vlan 200
NetIron(config-mpls-vpls-test_10-vlan-100-inner-vlan-200)#tagged Ethernet 2/1
Note – The system-max size for the Internal Forwarding Lookup CAM is 0.
Please use the command 'system-max ifl-cam' to specify a size.
```

  The informational message only warns that the configuration should be changed. It does not cause the system to reject the VPLS configuration. For example, in the sample case, the dual-tagged endpoint configuration of vlan 100 inner-vlan 200 on port ethernet 2/1 has been accepted assuming the port, outer VLAN, and inner VLAN combination has not already been assigned elsewhere.

NOTE
The acceptable values and configuration procedures for the **system-max ifl-cam** command are in the section "Configuring system max values" on page 106.

- When the IFL CAM partition on the Interface module exceeds a configured threshold, there will be a warning log message which is similar to the way other CAM partitions are handled currently. The system will not generate any logs if it cannot program the IFL CAM because of exhaustion of an IFL CAM resource.
- If an outer VLAN is specified for a given endpoint, it is called a less-specific VLAN. If both an outer VLAN and inner VLAN are specified, it is called a more-specific VLAN (in relation to the outer VLAN).
- Similar to single-tagged endpoints, the outgoing VLANs for a dual-tagged endpoint are based solely on the outgoing endpoint configuration, and not on the incoming packet VLAN values.

- The same port, outer VLAN, and inner VLAN combination cannot be specified across VPLS instances. For example, if a dual-tagged endpoint with VLAN 100 and inner VLAN 200 is configured on port ethernet 2/1 on VPLS instance “test”, same endpoint cannot be configured as part of another VPLS instance (for example, “test 1”).This is also true across applications. If a port, outer VLAN, and inner VLAN combination belongs to a VPLS instance, it cannot simultaneously belong to a Layer 2 VLAN, local VLL, or VLL.
- When CPU protection is enabled for a VPLS instance, the system will not support a configuration with two different dual-tagged VPLS VLANs as part of the same VPLS instance. Consider the following configuration example.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls test 10
NetIron(config-mpls)#cpu-protection
NetIron(config-mpls-vpls-test)#vlan 10 inner-vlan 20
NetIron(config-mpls-vpls-test-vlan-10-20)#tagged eth 2/1
NetIron(config-mpls-vpls-test-vlan-10)#exit
NetIron(config-mpls-vpls-test)#vlan 10 inner-vlan 30
NetIron(config-mpls-vpls-test-vlan-10-30)#tagged eth 2/1
Error - VPLS port 2/1 cannot be shared by multiple end-points when CPU
protection is enabled. Remove CPU protection for VPLS 10 to make this
configuration change.
```

  Similarly, CPU protection cannot be enabled for a VPLS instance that has a port configured under two different dual-tagged VPLS VLANs. Consider the following configuration example.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls test 10
NetIron(config-mpls-vpls-test)#vlan 10 inner-vlan 20
NetIron(config-mpls-vpls-test-vlan-10-20)#tagged eth 2/1
NetIron(config-mpls-vpls-test-vlan-10)#exit
NetIron(config-mpls-vpls-test)#vlan 30 inner-vlan 40
NetIron(config-mpls-vpls-test-vlan-30-40)#tagged eth 2/1
NetIron(config-mpls-vpls-test-vlan-20)#exit
NetIron(config-mpls-vpls-test)#cpu-protection
Error - Cannot configure CPU protection for VPLS 10 as multiple end-points
share the same physical port.
```

  The restrictions exist because packets are hardware-forwarded when CPU protection is enabled. In this case, source port suppression cannot be properly performed if there are multiple endpoints on the same physical interface.

Specifying an untagged endpoint

To specify an untagged endpoint for a VPLS instance, enter commands such as the following.

```
NetIron(config-mpls)# vpls v1 40000
NetIron(config-mpls-vpls-v1)# vlan 100
NetIron(config-mpls-vpls-v1-vlan-100)# untagged ethernet 2/1
```

Syntax: [no] untagged [ethernet] <*portnum*>

NOTE
Foundry Discovery Protocol (FDP) should not be enabled on an untagged VPLS or VLL endpoint.

Specifying a single-tagged endpoint

Tagged ports are configured under a VLAN ID. A VPLS instance can have multiple ports configured under the same VLAN ID, and can have ports configured under different VLAN IDs. Another VPLS instance can reuse the same VLAN ID on other physical ports. Because the VLANs are configured under different VPLS instances, they are different VPLS VLANs even though they use the same VLAN ID.

To specify a tagged endpoint for a VPLS instance, enter commands such as the following:

```
NetIron(config-mpls)# vpls v1 40000
NetIron(config-mpls-vpls-v1)# vlan 200
NetIron(config-mpls-vpls-v1-vlan-200)# tagged ethernet 3/11
```

Syntax: vlan <*num*>

Syntax: [no] tagged ethernet <*slot/port*>

Specifying a dual-tagged endpoint

Dual-tagged ports are configured with two VLAN IDs. A VPLS instance can have multiple ports configured under the same dual-tagged VLAN ID, and can have ports configured under different VLAN IDs. Another VPLS instance can reuse the same VLAN ID on other physical ports. Because the VLANs are configured under different VPLS instances, they are different VPLS VLANs even though they use the same VLAN ID.

NOTE
Before configuring a dual-tagged endpoint, see "Special considerations for dual-tagged endpoints" on page 1506.

To specify a dual-tagged endpoint for a VPLS instance, use the following commands.

```
NetIron(config-mpls)# vpls v1 40000
NetIron(config-mpls-vpls-v1)# vlan 200 inner-vlan 300
NetIron(config-mpls-vpls-v1-vlan-200)# tagged ethernet 3/11
```

Syntax: [no] vlan <*VLAN-ID*> **inner-vlan** <*VLAN-ID*>

Syntax: [no] tagged ethernet <*slot/port*>

The **vlan** <*VLAN-ID*> variable, which is the outer VLAN ID, can be in the range from 1 through 4094 and excludes the default VLAN.

The **inner-vlan** <*VLAN-ID*> variable, can be in the range from 1 through 4095 and includes the default VLAN.

Use the **no** form of the command to remove the dual-tagged VPLS VLAN configuration and its associated endpoints. For example, the command **no vlan 200 inner-vlan 300** will remove the dual-tagged VLAN and associated endpoints. The single-tagged VLAN, **vlan 200**, will not be deleted. Similarly, the command **no vlan 200** will remove the single-tagged VLAN, **vlan 200**, and associated endpoints. The dual-tagged VLAN, **vlan 200 inner-vlan 300**, will not be deleted.

Example of dual-tagged endpoints mapped to different VPLS instances

The following example shows two dual-tagged endpoints on the same physical interface with the same outer VLAN ID and different inner VLAN IDs mapped to different VPLS instances.

```
NetIron(config-mpls)#vpls test_10 10
NetIron(config-mpls-vpls-test_10)#vlan 100 inner-vlan 200
```

```
NetIron(config-mpls-vpls-test_10-vlan-100-inner-vlan-200)#tagged ethernet 2/1
NetIron(config-mpls-vpls-test_10-vlan-100-inner-vlan-200)#exit
NetIron(config-mpls-vpls-test_10)#exit
NetIron(config-mpls)#vpls test_20 20
NetIron(config-mpls-vpls-test_20)#vlan 100 inner-vlan 300
NetIron(config-mpls-vpls-test_20-vlan-100-inner-vlan-300)#tagged ethernet 2/1
NetIron(config-mpls-vpls-test_20-vlan-100-inner-vlan-300)#exit
```

Example of dual-tagged endpoints mapped to the same VPLS instance

The following example shows two dual-tagged endpoints on the same physical interface with the same outer VLAN ID and different inner VLAN IDs mapped to the same VPLS instance.

```
NetIron(config-mpls)#vpls test 10
NetIron(config-mpls-vpls-test)#vlan 100 inner-vlan 200
NetIron(config-mpls-vpls-test-vlan-100-inner-vlan-200)#tagged ethernet 2/1
NetIron(config-mpls-vpls-test-vlan-100-inner-vlan-200)#exit
NetIron(config-mpls-vpls-test)#vlan 100 inner-vlan 300
NetIron(config-mpls-vpls-test-vlan-100-inner-vlan-300)#tagged ethernet 2/1
```

In the above example, if packets are received with outer VLAN ID 100 and no inner VLAN ID, the packets will not be handled as part of VPLS instance 10.

Example of a less-specific and more-specific VLAN mapped to different VPLS instances

The following example shows a less-specific VLAN and more-specific VLAN (with the same outer VLAN ID) of the same port in different VPLS instances.

```
NetIron(config-mpls)#vpls test_10 10
NetIron(config-mpls-vpls-test_10)#vlan 100
NetIron(config-mpls-vpls-test_10-vlan-100)#tagged ethernet 2/1
NetIron(config-mpls-vpls-test_10-vlan-100)#exit
NetIron(config-mpls-vpls-test_10)#exit
NetIron(config-mpls)#vpls test_20 20
NetIron(config-mpls-vpls-test_20)#vlan 100 inner-vlan 300
NetIron(config-mpls-vpls-test_20-vlan-100-inner-vlan-300)#tagged ethernet 2/1
NetIron(config-mpls-vpls-test_20-vlan-100-inner-vlan-300)#exit
```

Example of a less-specific and more-specific VLAN mapped to the same VPLS instance

The following example shows a less-specific VLAN and more-specific VLAN (with the same outer VLAN ID) of the same port in the same VPLS instance.

```
NetIron(config-mpls)#vpls test_10 10
NetIron(config-mpls-vpls-test_10)#vlan 100
NetIron(config-mpls-vpls-test_10-vlan-100)#tagged ethernet 2/1
NetIron(config-mpls-vpls-test_10-vlan-100)#exit
NetIron(config-mpls-vpls-test_10)#vlan 100 inner-vlan 300
NetIron(config-mpls-vpls-test_10-vlan-100-inner-vlan-300)#tagged ethernet 2/1
NetIron(config-mpls-vpls-test_10-vlan-100-inner-vlan-300)#exit
```

In the above example, if packets are received on interface ethernet 2/1 with outer VLAN ID 100 and an inner VLAN ID other than 300, the packets will be handled as part of VPLS instance 10. In this case, the inner VLAN is treated as payload.

Specifying a LAG group as the endpoint of a VPLS instance

The endpoint of a VPLS instance can be a static or a dynamic LAG. When the endpoint of a VPLS instance is a LAG, the VPLS traffic load is distributed to the CE device across all of the LAG's ports by way of a hashing mechanism that utilizes the source and destination MAC addresses.

For example, to configure a LAG, enter commands such as the following.

```
NetIron(config)# lag blue dynamic
NetIron(config-lag-blue)# ports ethernet 1/1 to 1/2
NetIron(config-lag-blue)# primary-port 1/1
```

To configure a VPLS instance that uses the LAG defined as the endpoint by the previous example commands, enter the commands as in the following example.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls test1 40000
NetIron(config-mpls-vpls-test1)# vpls-peer 10.10.10.10
NetIron(config-mpls-vpls-test1)# vlan 200
NetIron(config-mpls-vpls-test1)# tagged e 1/1
```

NOTES: If you first create a LAG and then configure a VPLS instance, the port you specify as the VPLS endpoint must also be the port you specified as the primary port of the LAG group:

- If you first configure a VPLS instance and then create a LAG, all ports of the LAG must be specified as endpoints of the VPLS instance. The VPLS instance will use all the ports of the LAG.
- If you later delete the LAG from the configuration, all ports in the LAG become independent endpoints in the VPLS instance.
- If you specified a tagged endpoint for the VPLS instance, all of the ports in the LAG must be tagged.
- Traffic received from any port in the LAG is forwarded to the VPLS instance. All traffic is matched to its VLAN.

Support for VPLS endpoints within a Topology group

You can configure VPLS VLANs into Topology groups so that you can use any of the following protocols within a VPLS VLAN:

- Spanning Tree Protocol (STP)
- Rapid Spanning Tree Protocol (RSTP)
- Foundry Metro Ring Protocol (MRP)
- Virtual Switch Redundancy Protocol (VSRP)

Instructions for configuring VPLS VLANs into Topology groups are provided in the chapter 16, "Topology Groups" in the section titled "Adding VPLS VLANs to topology groups" on page 546.

Flooding Layer 2 BPDUs in VPLS

By default, Layer 2 STP and Per VLAN Spanning Tree (PVST) Bridge Protocol Data Units (BPDUs) entering a VPLS endpoint are not transparently flooded within the VPLS instance. The BPDUs are dropped when they enter the VPLS endpoint. The user can change this default behavior to not block BPDUs and transparently flood them within the VPLS instance, by configuration on a per-physical-port basis. Because the BPDU block option is configurable per physical interface, it will affect all VPLS instances that have endpoints on that interface.

To flood BPDUs in VPLS, use the following command.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# no vpls-bpdu-block
```

Syntax: [no] vpls-bpdu-block

Specifying the VPLS VC type

NOTE
This is a global setting that will affect all VPLS instances, except those in VPLS tagged mode. You must save the configuration and reload the software to place the change into effect.

The default VC type for all VPLS instances is set to 0x5 or “Ethernet”. For compatibility with previous versions, the VC type can be changed to 0xB or “Ethernet VPLS”. The VC type must match between peers for the VPLS session to be established.

NOTE
When VPLS tagged mode is enabled for a VPLS instance, the VC type for that instance will be set to 0x04 or “Ethernet Tagged”, regardless of the global VPLS VC type setting.

To change the VPLS VC type, use the following command at the MPLS configuration level.

```
NetIron(config-mpls)# vpls-vc-type-ethernet-vpls
VPLS VC type will be 0xB (Ethernet VPLS) after you save to config and reboot
```

Syntax: [no] vpls-vc-type-ethernet-vpls

Configuring VPLS tagged mode

This section describes how to enable, disable, and view the configuration details of VPLS tagged mode. For details about how VPLS tagged mode works, see “VPLS tagged mode” on page 1500.

Enabling VPLS tagged mode

To enable VPLS tagged mode, first create the VPLS instance if it does not already exist, and then enter commands such as the following at the MPLS VPLS configuration level of the CLI.

```
NetIron(config-mpls)#vpls test 100
NetIron(config-mpls-vpls-test)#vc-mode tagged
```

Syntax: vc-mode tagged

Disabling VPLS tagged mode

Except for VPLS instances on which ISID is configured, use the **no vc-mode tagged** command to disable VPLS tagged mode. For VPLS instances with an ISID configuration, first remove the ISID configuration, then disable VPLS tagged mode.

If you attempt to disable VPLS tagged mode on a VPLS instance with ISID, the system will display the following error message.

```
NetIron(config-mpls-vpls-test)#no vc-mode tagged
Error – Cannot remove tagged-mode setting while VPLS ISID configuration exists
```

Syntax: no vc-mode tagged

Viewing the VPLS tagged mode configuration

Use the **show running config** and **show mpls vpls detail** commands to view the VPLS tagged mode configuration. The following shows an example **show running config** output. For examples and details of the **show mpls vpls detail** command, see "Enabling MPLS VPLS traps" on page 1518.

```
NetIron(config-mpls)#vpls test 100
NetIron(config-mpls-vpls-test)#vc-mode tagged

NetIron#show running config
....
router mpls

vpls test 100
  vc-mode tagged
  vpls-peer 100.100.100.100
  vlan 100 inner-vlan 45
    tag e 2/1
vpls name_raw 3
  vpls-peer 200.200.200.200
```

In the above example, **vc-mode tagged** indicates that VPLS tagged mode is enabled on **vpls test 100**, whereas **vpls name_raw 3** is in VPLS raw mode.

VPLS CPU protection

The VPLS CPU protection feature protects the CPU of the line card from being overwhelmed by excessive VPLS packets that would require the CPU's attention, including unknown unicast, multicast packets, and packets requiring source-MAC learning. Once this feature is enabled, all VPLS multicast traffic will be hardware-flooded. Furthermore, when the CPU is too busy, this feature will hardware-flood unknown unicast traffic, as well as reduce the rate of source-MAC learning traffic to the line card CPU, so that the line card CPU will have enough resources to handle other types of packets.

Configuration Considerations

Note the following configuration rules before enabling VPLS CPU protection.

- VPLS CPU protection cannot be concurrently enabled with IGMP snooping on a VPLS instance.
- CPU protection cannot be enabled for a VPLS instance that has a port configured under two different VPLS VLANs. Similarly, when CPU protection is enabled for a VPLS instance, the system will not support a configuration with two different VPLS VLANs as part of the same VPLS instance.
- If VPLS FID usage reaches 100%, CPU protection will be temporarily disabled until adequate FID resources are available.

Configuring adequate CAM resources

NOTE
Beginning with release 03.8.00, the Multi-Service IronWare software has been enhanced to support the dynamic growth of the protocol and flooding sub-partitions in the L2 CAM to grant them a higher priority. Because of this enhancement, the **system-max hw-flooding** command is no longer required and has been retired from the software.

When using VPLS CPU protection, you must have adequate CAM resources available. Each endpoint and each uplink port requires a single CAM entry. Also, if an endpoint is a LAG port, one entry is required for each port in the LAG. To determine the number of entries required on your system, add the number of VPLS endpoints, ports within a LAG port used as an endpoint, and uplink ports. Use this number with the **system-max hw-flooding** command to provide adequate CAM resources.

By default, 8 CAM entries are available. To configure a larger number, you must use the following command.

```
NetIron(config)# system-max hw-flooding 40
```

Syntax: [no] system-max hw-flooding <*number*>

The <*number*> variable is the number of CAM entries that you want to make available.

NOTE
You must reload your system for the **system-max** command to take effect.

Configuring VPLS CPU protection

VPLS CPU protection can be configured in either of the following two ways:

- Globally – This enables VPLS CPU protection to affect all VPLS instances on the router.
- Per-VPLS – This enables VPLS CPU protection on one or more specified VPLS instances.

Configuring VPLS CPU protection globally

VPLS CPU protection can be enabled for all VPLS instances on a router. To enable VPLS CPU protection on all VPLS instances, enter the following command.

```
NetIron(config)# router mpls
NetIron(config-mpls) vpls-cpu-protection
```

Syntax: [no] vpls-cpu-protection

Configuring VPLS CPU protection per VPLS

VPLS CPU protection can be enabled per VPLS instance. To enable VPLS CPU protection on a specified VPLS instance, enter the following command.

```
NetIron(config)# router mpls
NetIron(config-mpls) vpls test 1
NetIron(config-mpls-vpls-test)# cpu-protection
```

Syntax: [no] cpu-protection

VPLS Broadcast, multicast, and unknown unicast packet limiting

This feature limits the number of broadcast, multicast, and unknown unicast packet packets that are flooded into the network. You can limit the number of broadcast, multicast, and unknown unicast packet that are flooded by the LP CPU on all VPLS instances.

You can limit the number of packets that are flooded by the LP CPU on VPLS instances using the following command.

```
NetIron(config)# router mpls
NetIron(config-mpls) vpls-policy
    cpu-broadcast-limit 100000
    cpu-multicast-limit 200000
    cpu-unknown-unicast-limit 300000
```

Syntax: [no] vpls-policy | [no] cpu-broadcast-limit <*number*> |[no] cpu-multicast-limit <*number*>| [no]cpu-unknown-unicast-limit <*number*>]

The <*number*> variable refers to the packet limits for broadcast, multicast, and unknown unicast packets.

CPU packet limiting

The CPU limiting feature affects the packets being flooded into VPLS endpoints as well as remote peers. The CPU packet limiting is as follows:

- When VPLS policy parameters are configured in the MP, the LP CPU acts on those configuration parameters and limits the number of packets of a specific type up to the configured limit.
- Once the number of packets for a given type exceeds the limit, the LP CPU drops the packets before flooding into the VPLS instance.
- When a packet is dropped, the drop-reason-code is incremented accordingly.
- Counters are reset and start counting the packets again after one second.
- Known unicast packets forwarded by the LP CPU will not be counted as drops as those packets will be sent directly to the next hop and not flooded. This can happen in CAM full scenarios.

Interaction with VPLS CPU protection

The interaction with VPLS CPU protection is as follows:

- When the VPLS CPU protection feature is enabled, multicast and broadcast packets are normally forwarded in hardware using a flooding entry after Source MAC Address (SA) learning. Most of the unicast packets will be forwarded in hardware as well, and packets will be sent to the CPU for flooding in a rate-limiting fashion depending upon CPU utilization.
- When VPLS CPU protection is configured along with the packet-limiting feature, the packets with CPU protection will be subjected to CPU limiting. In other words, the packets coming to the CPU for VPLS with CPU protection may be dropped with the CPU limiting feature enabled.

Multicast traffic processing

The multicast traffic processing is as follows:

- If a multicast packet limit is configured, multicast data packets will not be differentiated from multicast protocol packets.
- All the packets will be subjected to packet limitations and may be dropped if they exceed per-second multicast packet limits.
- Multicast snooping packets and IEEE 802.1ag packets may be affected as well.

Layer 2 control traffic behavior on VPLS endpoints

This section describes the layer 2 control traffic behavior on VPLS endpoints.

802.1x Protocol packets on a VPLS endpoint

802.1x does not support VPLS endpoints.

Cisco Discovery Protocol packets

Cisco Discovery Protocol (CDP) cannot be configured on a VPLS endpoint port and a VPLS endpoint cannot be configured on a physical port that has CDP enabled. This restriction is enforced by the CLI. If a VPLS endpoint receives any CDP traffic, this traffic will be transparently flooded within the VPLS.

The behavior of CDP control packets is as follows:

- If CDP is globally enabled on the device, and the **priority force** command is configured on an incoming port, the VPLS local switched packets will be sent out with a priority of 7.
- If CDP is not enabled on the device, packets are switched locally according to the priority in the configured **qos exp encode** command or **qos pcp encode-policy** command.

Foundry Discovery Protocol packets

Foundry Discovery Protocol (FDP) cannot be configured on a VPLS endpoint port and a VPLS endpoint cannot be configured on a physical port that has FDP enabled. This restriction is enforced by the CLI. If a VPLS endpoint receives any FDP traffic, this traffic will be transparently flooded within the VPLS.

The behavior of FDP control packets is as follows:

- If FDP is globally enabled on the device and **priority force** command is configured on an incoming port, the VPLS local switched packets will be sent out with a priority of 7.
- If FDP is not enabled on the device, packets are switched locally according to the priority in the configured **qos exp encode** command or **qos pcp encode-policy** command.

Uni-directional Link Detection packets

Uni-directional Link Detection (UDLD) cannot be configured on a VPLS endpoint port and a VPLS endpoint cannot be configured on a physical port that has UDLD enabled. This restriction is enforced by the CLI. If a VPLS endpoint receives any UDLD traffic, this traffic will be dropped by the router at ingress. However, if the VPLS has CPU protection enabled, this traffic will be hardware-flooded intermittently.

Flooding Layer 2 BPDUs with a VPLS instance

By default, Layer 2 Spanning Tree Protocol (STP) and Per VLAN Spanning Tree (PVST) BPDUs entering a VPLS endpoint are not transparently flooded within the VPLS instance. The BPDUs are dropped when they enter the VPLS endpoint. The user can change this default behavior to not block BPDUs and transparently flood them within the VPLS instance, by configuration on a per-physical-port basis.

To flood BPDUs with a VPLS, use the following command.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)#no vpls-bpdu-block
```

Syntax: [no] vpls-bpdu-block

Specifying a VPLS MTU

The **vpls-mtu** command allows you to specify an MTU value per VPLS instance. The newly configured VPLS MTU takes effect immediately to refresh or re-establish the VPLS sessions with peers in the following manner:

- If the VPLS session is Operational and the VPLS's MTU is changed by configuration, bring down the peer, we send a label withdraw message to the peer, followed by the current VC binding message.
- If the VPLS session is not Operational and the VPLS's MTU is changed by configuration and the current state of the peer is VC-Parameter-MTU-Mismatch, the peer is brought UP if the MTU is equal, and a VC withdraw message is sent to clean up the old binding on the peer, and then a VC binding message is sent with the newly configured MTU. If the current state of the peer is anything other than VC-Parameter-MTU-Mismatch, the VPLS's configured MTU is changed.
- When a VC binding is received from a peer and VPLS MTU enforcement is enabled, the received MTU is compared with the VPLS's MTU. if they are not equal, the peer is kept in the VC-Parameter-MTU-Mismatch state, and otherwise made Operational. If the MTU enforcement is disabled, the peer's MTU is saved and the peer is made Operational irrespective of the MTU values.

To configure a new MTU value for a VPLS instance, use the **vpls-mtu** command as shown in the following example.

```
NetIron(config-mpls)# vpls dell 40000
NetIron(config-mpls-vpls-dell)# vpls-mtu 1000
```

Syntax: [no] vpls-mtu <*mtu-value*>

The <*mtu-value*> variable can be set to any value between 64 through 9190.

If you use the **no** parameter to remove the configured MTU value for a VPLS instance, that instance's MTU becomes one of two possible values:

- If a global default MTU has been configured, then the MTU for this VPLS instance becomes that global maximum frame size minus 26.
- If no global default exists, the MTU is 1500.

For example, if you remove the MTU value for the VPLS named Dell, and the global default maximum frame size is 5000, then the MTU for VPLS Dell becomes 4974 (5000 – 26 = 4974).

NOTE
This MTU parameter is not enforced on the data plane (hardware). Consequently, packets larger than the configured MTU can still be sent or received.

Configuring VPLS MTU enforcement

You can set the device to enforce the VPLS MTU value when establishing control sessions with peers. This is done globally on the router using the **vpls-mtu-enforcement** command.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls-mtu-enforcement
```

Syntax: [no] vpls-mtu-enforcement

NOTE
The **vpls-mtu-enforcement** command is global to all VPLS instances. It requires a reload to take effect.

Configuring VPLS local switching

VPLS local switching is enabled by default, so packets received on a VPLS endpoint are flooded or forwarded to other VPLS endpoints belonging to the VPLS instance. This mode of operation does not require any configuration.

Using the **no vpls-local-switching** command, you can disable VPLS local switching. With VPLS local switching disabled, packets will only be flooded to the VPLS peers in a VPLS instance and not to the other VPLS endpoints belonging to that instance. Also, unicast traffic will be discarded if it is received on a VPLS endpoint and is meant to go out on another VPLS endpoint.

You can disable VPLS local switching behavior on a per-VPLS basis using the **no vpls-local-switching** command.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls test 100
NetIron(config-mpls-vpls-test)# no vpls-local-switching
```

Syntax: [no] vpls-local-switching

Once the **no vpls-local-switching** command has been used to disable VPLS local switching, you can use the command without the **no** option to turn VPLS local switching on.

Special considerations

When using the VPLS local switching feature, consider the following:

- When you toggle this option, all the MAC addresses that were learned on the VPLS endpoints are flushed and re-learned.
- This option does not affect IGMP/PIM snooping. Multicast traffic continues forwarding only to those VPLS endpoints and peers from which a Join for the (S,G) is requested, regardless of the status of the local switching option.
- IEEE 802.1ag packets will follow the local switching option. In other words, packets are forwarded or flooded to other VPLS endpoints if local switching is enabled and discarded if local switching is disabled.

Enabling MPLS VPLS traps

You can enable traps that will be generated for MPLS VPLS by entering the following command.

```
NetIron(config)# snmp-server enable trap mpls vpls
```

Syntax: [no] snmp-server enable trap mpls vpls

Refer to the *IronWare MIB Reference* for MPLS VPLS trap notifications.

Disabling Syslog messages for MPLS VPLS

The generation of Syslog messages for MPLS VPLS and MPLS VLL Local is enabled by default. If you want to disable the logging of these events, enter the following command.

```
NetIron(config)# no logging enable mpls
```

Syntax: [no] logging enable mpls

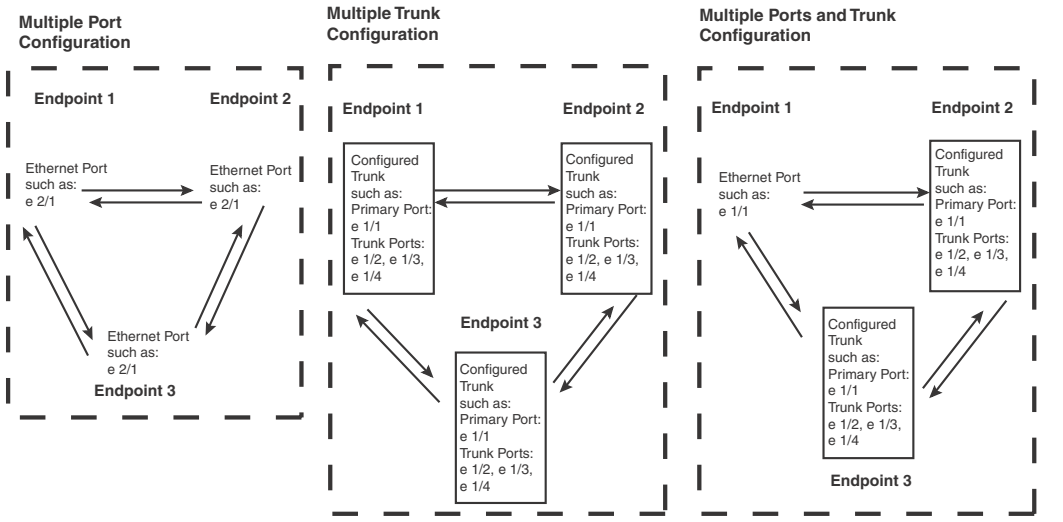
Refer to Table 454 on page 2299 for a list of Syslog messages for MPLS.

Local VPLS

Local VPLS is used to create a VPLS circuit with endpoints in the same device. A Local VPLS can be configured between two or more ports in a router, two or more LAGs in a router, or between a port and a LAG as shown in Figure 199. Each entity (port or LAG) is identified as “Endpoint 1”, Endpoint 2” or “Endpoint 3”.

NOTE
Trunks supported include server LAGs and per-packet server LAGs. Link Aggregation Control Protocol (LACP) LAGs are not supported.

FIGURE 199 Local VPLS port and LAG configurations



Note: In this configuration, any endpoint can configured as either a trunk or a single port.

NOTE
When configuring a LAG as an endpoint, only the primary port of the LAG is specified in the Local VPLS configuration.

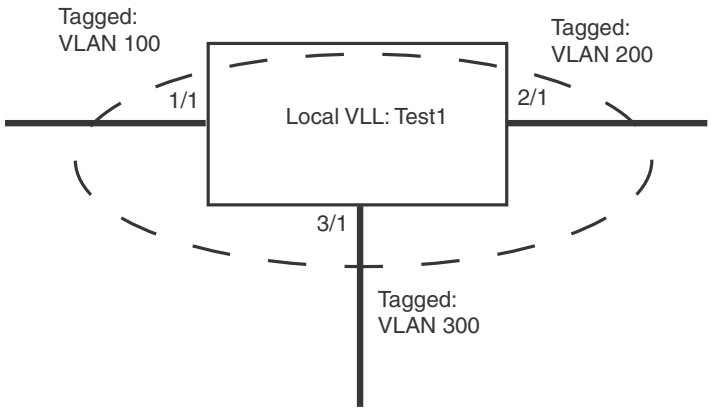
NOTE
Packets that arrive on an interface with the same destination MAC address as the interface are forwarded in hardware just like packets with other destination addresses.

The endpoints connected to the Local VPLS can be untagged, dual tagged, or single-tagged as members of the same or different VLANs. Using this function of Local VPLS, a router can receive packets with particular tags or no tag on one endpoint and forward them to the Local VPLS's other endpoint, which may be untagged, dual-tagged, or single-tagged with a different VLAN tag. When so configured, the tags within the packets are changed to reflect the configuration of the egress port as they leave the router.

Example Local VPLS configuration

In Figure 200 the Local VPLS named "Test1" contains Ethernet ports 1/1, 2/1, and 3/1. Port 1/1 is a member of VLAN 100, port 2/1 is a member of VLAN 200, and port 3/1 is a member of VLAN 300. Because all of the ports belong to Local VPLS "Test1", traffic tagged with any of the configured tags (100, 200, or 300) can reach traffic within any of the three VLANs. For example, traffic that ingresses on port 1/1 must have a tag with the value "100" and will egress on port 2/1 with a tag value of "200" or egress on port 3/1 with a tag value of "300".

FIGURE 200 Local VPLS "Test1" with three tagged VLANs




```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls-test1 5000
NetIron(config-mpls-vpls-test1)# vlan 100
NetIron(config-mpls-vpls-test1-vlan-100)# tagged ethernet 1/1
NetIron(config-mpls-vpls-test1-vlan-100)# vlan 200
NetIron(config-mpls-vpls-test1-vlan-200)# tagged ethernet 2/1
NetIron(config-mpls-vpls-test1-vlan-200)# vlan 300
NetIron(config-mpls-vpls-test1-vlan-300)# tagged ethernet 3/1
```

CoS behavior for Local VPLS

NOTE
This section assumes that you understand how QoS works. For details, see "Configuring Quality of Service for the NetIron MLX" on page 283.

Table 254 describes the expected Class of Service (CoS) behavior for VPLS packets when Local VPLS is in effect.

TABLE 254 Expected class of service behavior for Local VPLS

Local VPLS endpoints	Incoming packet		Outgoing packet	
	Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN
Dual-tagged to dual-tagged	X	Y	X' or X	Y
Single-tagged to dual-tagged	X	N/A	X' or X	X
Untagged to dual-tagged	N/A	N/A	X' or 0	0
Dual-tagged to single-tagged	X	Y	X' or Y	N/A

Legend for Table 254

X = Original outer VLAN CoS.

Y = Original inner VLAN CoS.

X= Mapped CoS from internal priority (X contributes to internal priority) using CoS encode table.

Specifying Local VPLS endpoints

Local VPLS can be configured between any combination of dual-tagged, single-tagged, and untagged endpoints.

The following procedures describe how to configure VPLS endpoints:

- "Configuring an untagged endpoint"
- "Configuring a single-tagged endpoint"
- "Configuring a dual-tagged endpoint"

Configuring an untagged endpoint

To configure untagged port 1/1 into Local VPLS instance "test1", use the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls test1 5000
NetIron(config-mpls-vpls-test1)# untagged ethernet 1/1
```

Syntax: [no] untagged ethernet <*slot/port*> <*vpls-id*>

The <*vpls-id*> variable variable is the ID of a VPLS instance.

Configuring a single-tagged endpoint

Tagged ports are configured under a VLAN ID. This VLAN ID is only meaningful for the tagged port.

For tagged ports, a <*vlan-id, port*> variable pair constitutes a VPLS endpoint. If a port is currently a member of a non-default VLAN as an untagged port, it must be returned to the default VLAN before it can be assigned to a VPLS as a tagged port.

To configure a tagged port 1/2 with VLAN 200 into Local VPLS instance "test1", use the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls test1
NetIron(config-mpls-vpls-test1)# vlan 200
NetIron(config-mpls-vpls-test1-vlan-200)# tagged ethernet 1/2
```

Syntax: vlan <*VLAN-ID*>

The range for <*VLAN ID*> from 1 through 4094. (This parameter range excludes the default VLAN ID.)

Syntax: [no] tagged ethernet <*slot/port*>

The <*slot/port*> variable specifies the port that is a tagged ethernet port.

Configuring a dual-tagged endpoint

A dual-tagged endpoint enables packets to have both an outer VLAN tag and an inner VLAN tag. In this configuration, an endpoint can receive packets with two tags and forward them to the other endpoint either untagged, single-tagged, or dual-tagged.

NOTE
Dual-tagged endpoints for Local VPLS follow the same configuration rules as do endpoints of a VPLS instance. Before configuring a dual-tagged endpoint, see "Special considerations for dual-tagged endpoints" on page 1506.

To configure a dual-tagged endpoint for Local VPLS, use the following commands.

```
NetIron(config)# router mpls
NetIron(config-mpls)# vpls test1
NetIron(config-mpls-vpls-test1)# vlan 200 inner-vlan 300
NetIron(config-mpls-vpls-test1-vlan-200)# tagged ethernet 1/2
```

Syntax: [no] vlan <*VLAN-ID*> **inner-vlan** <*VLAN-ID*>

Syntax: [no] tagged ethernet <*slot/port*>

The **vlan** <*VLAN-ID*> variable, which is the outer VLAN ID, can be in the range from 1 through 4094 and excludes the default VLAN ID.

The **inner-vlan** <*VLAN-ID*> variable can be in the range from 1 through 4095 and includes the default VLAN ID.

Use the **no** form of the command to remove the dual-tagged VPLS VLAN configuration and its associated endpoints. For example, the command **no vlan 200 inner-vlan 300** will remove the dual-tagged VLAN and associated endpoints. The single-tagged VLAN, **vlan 200**, will not be deleted. Similarly, the command **no vlan 200** will remove the single-tagged VLAN, **vlan 200**, and associated endpoints. The dual-tagged VLAN, **vlan 200 inner-vlan 300,** will not be deleted.

Displaying VPLS information

You can display the following information about the VPLS configuration on the device:

- VPLS summary information
- Information about individual VPLS instances configured on the device
- Detailed information about VPLS instances
- Information about a specified VPLS ID or VPLS name
- Information about VPLS instances that are not fully operational
- The contents of the VPLS MAC database for a VPLS instance
- The VPLS MAC database entries on the Managment Processor (MP)
- VPLS traffic statistics
- VPLS CPU protection configuration status

Display considerations for VPLS information

The VPLS information that is displayed in the output of the **show mpls vpls** commands has changed. Previously, when a VPLS was created, a range of VC labels was allocated to the VPLS instance. Now, there is no pre-allocation of VC label ranges to a VPLS instance.

NOTE
This change is supported on PowerConnect B-MLXe devices.

The range of the allocated VC labels is no longer displayed in the output of the following **show mpls vpls** commands. Refer to the subsequent sections for more information on changes to the **show mpls vpls** command outputs:

- **show mpls vpls brief** - "Displaying information about VPLS instances" on page 1523
- **show mpls vpls detail** - "Displaying detailed information about VPLS instances" on page 1524
- **show mpls vpls down** - "Displaying information about VPLS instances that are not operational" on page 1531
- **show mpls vpls id** - "Displaying information about a specified VPLS ID or VPLS name" on page 1528
- **show mpls vpls summary** - "Displaying VPLS summary information" on page 1523

Displaying VPLS summary information

The **show mpls vpls summary** command has changed. The VC label allocation range size field is no longer displayed in the output of the **show mpls vpls summary** command.

You can display a summary of VPLS information, including the number of VPLS instances, number of VPLS peers, maximum size of the VPLS MAC database, VPLS raw mode, and the values of the VPLS global MTU, and the value of the remote VC MTU.

```
NetIron# show mpls vpls summary
Virtual Private LAN Service summary:
Total VPLS configured: 3, maximum number of VPLS allowed: 4096
Total VPLS peers configured: 1, total peers operational: 1
Maximum VPLS macentries allowed: 8192, currently installed: 3
VPLS global raw mode VC-Type is Ethernet (0x5)
VPLS global MTU is 1500, MTU enforcement is OFF
Global CPU protection: OFF
MVIDs in use: 1 of 1 total allocated
```

Syntax: show mpls vpls summary

Displaying information about VPLS instances

The **show mpls vpls brief** command has changed. The Num VC-label field is no longer displayed in the output of the **show mpls vpls brief** command.

To display information about VPLS instances configured on the device, enter the following command.

```
NetIron#show mpls vpls brief

                        Num   Num     Ports   Num    Peers          CPU  VC
Name          Id        Vlans Ports   Up      Peers  Up     IFL-ID  Prot Mode
====          ==        ===== =====   =====   =====  =====  ======  ==== ======
1             1         2     2       2       1      1      4096    OFF  TAGGED
2             2         1     0       0       1      0       n/a    OFF  RAW
3             3         2     6       4       2      1       n/a    OFF  RAW
```

Syntax: show mpls vpls brief

Table 255 lists the output displayed by the **show mpls vpls brief** command.

TABLE 255 Output from the **show mpls vpls brief** command

Field	Description
Name	The configured name of the VPLS instance.
Id	The ID of this VPLS instance.
Num Vlans	The total number of single-tagged and dual-tagged VLANs associated with this VPLS instance.
Num Ports	The number of ports in this VPLS instance.
Ports Up	The number of ports in this VPLS instance that are up.
Num Peers	The number of VPLS peers this device has for this VPLS instance.
Peers Up	The number of VPLS peers with which a VC connection is completely operational.
IFL-ID	The Internal Forwarding Lookup Identifier (IFL-ID) for dual-tagged VLAN ports in this VPLS instance.
CPU Prot	Whether CPU protection configured on this VPLS instance is ON or OFF.
VC Mode	The VC mode for the VPLS instance: • Raw – The VLAN tag information in the original payload is not carried across the MPLS cloud. • Tagged – The VLAN tag information in the original payload is carried across the MPLS cloud.

Displaying detailed information about VPLS instances

The **show mpls vpls detail** command has changed. The total VC labels allocated field is no longer displayed in the output of the **show mpls vpls detail** command.

To display more detailed information about each VPLS instance, enter the following command.

```
NetIron#show mpls vpls detail
VPLS 3, Id 3, Max mac entries: 8192
 Total vlans: 2, Tagged ports: 2 (1 Up), Untagged ports 0 (0 Up)
 IFL-ID: n/a
  Vlan 500
   Tagged: ethe 1/3
  Vlan 600
   Tagged: ethe 1/4
 VC-Mode: Raw
 Total VPLS peers: 1 (1 Operational)
 Peer address: 21.21.21.21, State: Operational, Uptime: 1 min
  Tnnl in use: tnl0(3)
  LDP session: Up, Local VC lbl: 983040, Remote VC lbl: 983040
  Local VC MTU: 1500, Remote VC MTU: 9174
  Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
 CPU-Protection: OFF [Resource FID Failure, Retry in 18 seconds (approximate)]
 Local Switching: Enabled
 Mutlicast Snooping: Disabled
```

Syntax: show mpls vpls detail

Table 256 lists the output displayed by the **show mpls vpls detail** command.

TABLE 256 Output from the **show mpls vpls detail** command

Field	Description
VPLS	The configured name of the VPLS instance.
Id	The ID of this VPLS instance.
Max mac entries	The maximum number of MAC address entries that can be learned for this VPLS instance. This is a soft limit only and can be exceeded if there is space available in the VPLS MAC database.
Total vlans	The number of VLANs that are translated for this VPLS instance.
Tagged ports	The total number of tagged ports that are associated with VLANs in this VPLS instance, as well as the number of these ports that are up.
Untagged ports	The total number of untagged ports that are associated with VLANs in this VPLS instance, as well as the number of these ports that are up.
IFL-ID	The Internal Forwarding Lookup Identifier (IFL-ID) for dual-tagged ports in the VPLS instance.
Vlan	The ID of each VLAN in this VPLS instance.
Tagged	The numbers of the tagged ports in each VLAN.
Untagged	The numbers of the untagged ports in each VLAN.
VC-Mode	The VC mode for the VPLS instance: • Raw – The VLAN tag information in the original payload is not carried across the MPLS cloud. • Tagged – The VLAN tag information in the original payload is carried across the MPLS cloud.
Total VPLS peers	The number of VPLS peers this device has for this VPLS instance, as well as the number of these VPLS peers with which this device has an LDP session.
Peer address	The IP address of the VPLS peer.

TABLE 256 Output from the **show mpls vpls detail** command (Continued)

Field	Description
State	The current state of the connection with the VPLS peer. This can be one of the following states: • Operational – The VPLS instance is operational. Packets can flow between the device and the peer. • Wait for functional local ports – The physical endpoint port that should be connected to the Customer Edge device is down due to a link outage or is administratively disabled. • Wait for LSP tunnel to Peer – The device cannot find a working tunnel LSP. • Wait for PW Up (Wait for LDP session to Peer)– The LDP session is not yet ready. • Wait for PW Up (Wait for remote VC label) - The device has advertised its VC label binding to the VPLS peer, but has not yet received the peer's VC label binding. • Wait for PW Up (VC type mismatched) – A session is not formed because the VC type does not match with its peer's VC type. • Wait for PW Up (MTU mismatched) – The MTU sent to a peer is derived from the router's global setting by the following formula: (*system-mtu* minus 26 bytes). If a *system-mtu* value is not configured, a default value of 1500 is sent. • Wait for PW Up (Wait for LPD session to Peer) - The LDP session to the peer is down. • Wait for PW Up (No Label Resource) - When configuring a new VPLS peer, the maximum amount of VC labels that can be supported may exceed 64K, and cause the configuration to be rejected. The maximum amount of VC labels available for VPLS instances is equal to 64K.
Uptime	The time in minutes that the entry has been operational.
Tnnl in use	The tunnel LSP used to reach the VPLS peer. If VPLS traffic to the peer is load balanced across multiple tunnel LSPs, the tunnel LSPs used to reach the peer are displayed.
LDP session	The state of the LDP session between this device and the VPLS peer.
Local VC lbl	The VC label value locally allocated for this peer for this VPLS instance. Packets forwarded from the VPLS peer to this device are expected to contain this label. This is the label that is advertised to the VPLS peer through LDP.
Remote VC lbl	The VC label allocated by the VPLS peer and advertised to this device through LDP. The device applies this label to outbound MPLS packets sent to the VPLS peer.
Local VC MTU	The MTU value locally configured for this peer.
Remote VC MTU	The MTU value configured for the remote VPLS peer.
Local VC-Type	The VC type for this peer.
Remote VC-Type	The VC type for the remote VPLS peer.
CPU-Protection	Whether CPU protection configured on this VPLS instance is ON or OFF.
Local Switching	Whether local switching behavior on a per-VPLS basis is enabled or disabled.

The Wait for LDP session to Peer state is no longer displayed in the output of the **show mpls vpls detail** command. The Wait for Pseudo Wire (PW) Up (Wait for LDP session to Peer) state is now displayed, and replaces the existing state. The total VC labels allocated field is also removed from the output. In the following example, the LDP session to the remote peer is down. The Local VC lbl field will displays N/A (not applicable).

```
NetIron#show mpls vpls detail
VPLS NO_LDP, Id 500, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4101
  Vlan 880 inner-vlan 35
   Tagged: ethe 8/2
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
Peer address: 66.66.66.66, State: Wait for PW Up (Wait for LDP session to Peer)
  Tnnl in use: tnl5(6)
  LDP session: Down, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0,
  LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
 CPU-Protection: OFF
 Local Switching: Enabled
```

The maximum number of VC labels available for VPLS instances is equal to 64K. When configuring a new VPLS peer, the total number of VPLS peers will exceed 64K, and will cause the configuration to be rejected. The following error message is displayed on the console.

```
NetIron(config-mpls-vpls-1)#vpls-peer 23.23.23.23
Error - Unable to create vpls peer 23.23.23.23 for VPLS 1 due to no VC label
resource.
```

The Wait for PW Up (No label Resource) state is introduced in the output of the **show mpls vpls detail** command. In the following example, the Wait for PW Up (No label Resource) state is highlighted.

```
NetIron#show mpls vpls detail
VPLS waiting_for_remote_label, Id 400, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4100
  Vlan 900 inner-vlan 245
   Tagged: ethe 7/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 55.55.55.55, State: Wait for PW Up (No Label Resource)
  Tnnl in use: tnl4(5)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0,
  LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
 CPU-Protection: OFF
 Local Switching: Enabled
```

When the system runs out of memory, a warning message is displayed on the console. To recover from this state, the user is required to delete the failed peer and reconfigure it. VPLS generates the following warning messages.

```
WARNING: VPLS id 3 Peer IP Address: 21.21.21.21 is placed in VC Bind Failure state
due to low system memory.
```

```
WARNING: VPLS id 3 Peer IP Address: 11.11.11.11  is placed in VC Withdraw Failure
state due to low system memory.
```

Displaying information about a specified VPLS ID or VPLS name

The **show mpls vpls id** <*vpls-id*> command displays detailed information about a specified VPLS ID. The **show mpls vpls name** <*vpls-name*> command displays detailed information about a VPLS name. The output of the **show mpls vpls id** <*vpls-id*> command, and the output of the **show mpls vpls name** <*vpls-name*> command display the same information for a configured VPLS instance. The display changes that are described below are applicable to both the **show mpls vpls id** <*vpls-id*> command, and **show mpls vpls name** <*vpls-name*> command.

When the remote peer is in an operational state, the total VC labels allocated field is no longer displayed in the output of the **show mpls vpls id** <*vpls-id*> command, as shown in the following example.

```
NetIron#show mpls vpls id 3
VPLS name_raw, Id 3, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 3 (3 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4097
  Vlan 300 inner-vlan 500
   Tagged: ethe 3/1 ethe 3/11 ethe 3/13
 VC-Mode: Raw
 Total VPLS peers: 1 (1 Operational)
 Peer address: 200.200.200.200, State: Operational, Uptime: 1 hr 10 min
  Tnnl in use: tnl1(4)
  LDP session: Up, Local VC lbl: 983072, Remote VC lbl: 983072
  Local VC MTU: 1500, Remote VC MTU: 1500
  LOCAL VC-Type: Ethernet (0x05), Remote VC-Type: Ethernet (0x05)
 CPU-Protection: OFF
 Local Switching: Enabled
```

When a VC type mismatch occurs, the output from the **show mpls vpls id** <*vpls-id*>command will now display the Wait for PW Up (VC type mismatched) state. The Wait for VC parameter check (VC type mismatched) state is no longer displayed. The total VC labels allocated field is also removed from the output. In the following example, a VC type mismatch has occurred, and the PW is down. The local VC MTU and the remote VC MTU are not known by VPLS so there is no information to display. The Local VC lbl field, the Remote VC lbl field, and the Remote VC MTU field display N/A (non applicable).

```
NetIron#show mpls vpls id 200
VPLS vc_mismatched, Id 200, Max macentries: 8192
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
IFL-ID: 4098
Vlan200 inner-vlan145
Tagged: ethe2/1
VC-Mode: Tagged
Total VPLS peers: 1 (0 Operational)
Peer address: 33.33.33.33, State: Wait for PW Up (VC type mismatched)
Tnnlin use: tnl0(2)
LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
Local VC MTU: 1500, Remote VC MTU: N/A
LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: Ethernet (0x05)
```

When a MTU mismatch occurs, the output from the **show mpls vpls id** <*vpls-id*> command will now display the Wait for PW Up (MTU mismatched) state. The Wait for VC parameter check (MTU mismatched) state is no longer displayed. The total VC labels allocated field is also removed from the output. In the following example, a MTU mismatch has occurred, and the PW is down. The Local VC lbl field and the Remote VC lbl field display N/A (not applicable).

NOTE
When both the VC type and MTU are mismatched, only the output from the VC type mismatch is displayed on the console.

```
NetIron#show mpls vpls id 300
VPLS mtu_mismatched, Id 300, Max macentries: 8192
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
IFL-ID: 4099
Vlan100 inner-vlan145
Tagged: ethe1/1
VC-Mode: Tagged
Total VPLS peers: 1 (0 Operational)
Peer address: 44.44.44.44, State: Wait for PW Up (MTU mismatched)
Tnnlin use: tnl3(3)
LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
Local VC MTU: 1500, Remote VC MTU: 2500,
LOCAL VC-Type: Ethernet Tagged (0x04), Ethernet Tagged (0x04)
```

The Wait for remote VC label from Peer state is no longer displayed in the output of the **show mpls vpls id** <*vpls-id*> command. The Wait for PW Up (Wait for remote VC label) state is now displayed, and replaces the existing state. The total VC labels allocated field is also removed from the output. In the following example, the PW is down and it is waiting for the VC label of the remote peer to be advertised to the VPLS peer. The Local VC lbl field and the Remote VC MTU field will display N/A (non applicable).

```
NetIron#show mpls vpls id 400
VPLS waiting_for_remote_label, Id 400, Max macentries: 8192
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
IFL-ID: 4100
Vlan900 inner-vlan245
Tagged: ethe7/1
VC-Mode: Tagged
Total VPLS peers: 1 (0 Operational)
Peer address: 55.55.55.55, State: Wait for PW Up (Wait for remote VC label)
Tnnlin use: tnl4(5)
LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
Local VC MTU: 1500, Remote VC MTU: N/A,
LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
```

The Wait for PW Up (VC Bind in Progress) state is introduced in the output of the **show mpls vpls id** <*vpls-id*> command. The total VC labels allocated field is removed from the output. In the following example, the PW is down, and local VC binding is still in progress. The Local VC lbl field and the Remote VC MTU field display N/A (non applicable).

```
NetIron#show mpls vpls id 400
VPLS waiting_for_remote_label, Id 400, Max macentries: 8192
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
IFL-ID: 4100
Vlan900 inner-vlan245
Tagged: ethe7/1
VC-Mode: Tagged
Total VPLS peers: 1 (0 Operational)
Peer address: 55.55.55.55, State: Wait for PW Up (VC Bind in Progress)
Tnnlin use: tnl4(5)
LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
Local VC MTU: 1500, Remote VC MTU: N/A,
LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
```

The **show mpls vpls id** <*vpls-id*> command displays the tunnel LSPs that are being used to forward VPLS traffic from the device to the peer. If VPLS traffic to a peer is being load balanced across multiple tunnel LSPs, then the command lists the tunnel LSPs used for load balancing, as shown in the example below.

```
NetIron# show mpls vpls id 5
VPLS test5, Id 5, Max mac entries: 2048
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  Vlan 50
   Tagged: ethe 5/3
 VC-Mode: Raw
 Total VPLS peers: 1 (1 Operational)
 Peer address: 5.5.5.5, State: Operational, Uptime: 28 min
  Tnnl (load balance): tnl0(3) tnl2(3) tnl1(3)
  LDP session: Up, Local VC lbl: 983040, Remote VC lbl: 983040
```

Syntax: show mpls vpls id <*vpls-id*>

Syntax: show mpls vpls name <*vpls-name*>

The <*vpls-id*> variable is the ID of a VPLS instance. The <*vpls-name*> variable is the name of a VPLS instance.

Displaying VPLS CPU protection configuration status

The **show mpls vpls id** command has changed. The total VC labels allocated field is no longer displayed in the output of the **show mpls vpls id** command.

To see the VPLS CPU protection configuration status for a specified VPLS, use the **show mpls vpls id** command.

```
NetIron(config)# show mpls vpls id 1
VPLS test1, Id 1, Max mac entries: 2048
 Total vlans: 1, Tagged ports: 1 (0 Up), Untagged ports 1 (1 Up)
  Vlan 2
   Tagged: ethe 5/4
   Untagged: ethe 2/2
 Total VPLS peers: 1 (0 Operational)
 Peer address: 1.1.1.1, State: Wait for remote VC label from Peer
  Tnnl: tnl0(3), LDP session: Up, Local VC lbl: 983040, Remote VC lbl: N/A
 CPU-Protection: ON,  MVID: 0x000,  VPLS FID: 0x00000205
```

The CPU protection status is highlighted in the previous example. It can be either on or off. If CPU protection status is enabled on the VPLS but is temporarily off due to unavailable FID resources, the following message is shown in the CPU-Protection field:

```
CPU-Protection: OFF [Resource FID Failure, Retry in 18 seconds (approximate)]
```

Syntax: show mpls vpls id [<*vpls-id*>]

The <*vpls-id*> variable is the ID of a VPLS instance.

Displaying information about VPLS instances that are not operational

The **show mpls vpls down** command has changed. The Num VC-label field is no longer displayed in the output of the **show mpls vpls down** command.

To display information about VPLS instances that are not fully operational, enter the following command.

```
NetIron#show mpls vpls down
The following VPLS'es are not completely operational:
                      Num     Num     Ports   Num     Peers
Name          Id      Vlans   Ports   Up      Peers   Up
test1         1       1       1       1       1       0
test2         2       1       2       1       1       0
test3         3       1       1       1       1       0
test4         4       1       2       1       1       0
```

Syntax: show mpls vpls down

Displaying the contents of the VPLS MAC database

The VPLS MAC database stores entries associating remote MAC addresses with VC LSPs and local MAC addresses with CE devices. When a PE device receives a Layer 2 frame from an attached CE device with a given destination MAC address, the PE device looks up the MAC address in the VPLS MAC database and assigns the frame to the associated VC LSP. Each VPLS instance configured on the PE device has a separate VPLS MAC database.

Displaying VPLS MAC database entries on the Management Processor

To display the entire VPLS MAC database on the Management Processor (MP), enter the following command.

```
NetIron# show mac vpls
Total VPLS mac entries in the table: 10 (Local: 5, Remote: 5)

                              Vlan:Inner-Vlan
VPLS MAC Address    L/R Port      /Peer       Age
==== ===========    === ====  =========       ===
1    0016.0100.1601 R   5/1   3.3.3.3         0
1    0010.0100.1003 L   5/3   2               0
1    0016.0100.1603 R   5/1   3.3.3.3         0
1    0010.0100.1005 L   5/3   2               0
1    0010.0100.1002 L   5/3   2               0
1    0016.0100.1605 R   5/1   3.3.3.3         0
1    0016.0100.1602 R   5/1   3.3.3.3         0
1    0010.0100.1004 L   5/3   2               0
1    0010.0100.1001 L   5/3   2               0
1    0016.0100.1604 R   5/1   3.3.3.3         0
1    0000.0201.0201 L   5/4   100:200         0
```

If a given remote VPLS MAC address is learned on multiple uplink interfaces, the Port field in the output of the **show mac vpls** command indicates “Mult.” instead of a port number. For example, this abbreviation might appear when all of the following are true:

- The remote PE establishes multiple LSPs to this router.
- Packets from a remote VPLS MAC address are load balanced across these LSPs.
- The packets arrive on different MPLS uplink interfaces at this router.

```
NetIron#show mac vpls
Total VPLS mac entries in the table: 2274 (Local: 8, Remote: 2266)

                                 Vlan:Inner-Vlan
VPLS  MAC Address     L/R Port      /Peer       Age
====  ===========     === ====  =========       ===
3     0012.f29b.d419  L   4/2   3               0
504   0060.2d00.0067  R   Mult. 10.99.42.253    0
504   00c0.f033.b24c  R   Mult. 10.99.42.253    0
504   0080.ad73.6185  R   1/1   10.99.42.253    375
504   00e0.c900.40cf  R   Mult. 10.99.42.253    0
504   0015.7719.d7f4  R   Mult. 10.99.42.253    0
504   00c0.f044.d58b  R   Mult. 10.99.42.253    0
504   0010.5a5c.5a3b  R   Mult. 10.99.42.253    0
504   00c0.f044.d696  R   Mult. 10.99.42.253    0
```

To see details for all the ports on which a remote VPLS MAC address has been learned, use the **show mac mpls vpls** <*mac-address*> command.

To display the VPLS MAC database on the MP for a VPLS instance specified by its VPLS ID, enter the following command.

```
NetIron# show mac vpls 1
Total MAC entries for VPLS 1: 10 (Local: 5, Remote: 5)

                                Vlan:Inner-Vlan
VPLS MAC Address    L/R Port      /Peer       Age
==== ===========    === ====  =========       ===
1    0016.0100.1601 R   5/1   3.3.3.3         0
1    0010.0100.1003 L   5/3   2               0
1    0016.0100.1603 R   5/1   3.3.3.3         0
1    0010.0100.1005 L   5/3   2               0
1    0010.0100.1002 L   5/3   2               0
1    0016.0100.1605 R   5/1   3.3.3.3         0
1    0016.0100.1602 R   5/1   3.3.3.3         0
1    0010.0100.1004 L   5/3   2               0
1    0010.0100.1001 L   5/3   2               0
1    0016.0100.1604 R   5/1   3.3.3.3         0
1    0000.0201.0201 L   5/4   100:200         0
```

To display a specific entry in the VPLS MAC database on the MP, enter the following command.

```
NetIron# show mac vpls 1 0016.0100.1601
    VPLS: 1         MAC:  0016.0100.1601     Age:     0
    Remote MAC      Port: ethe 5/1           Peer:    3.3.3.3
Trunk slot mask: 00000000
```

Syntax: show mac vpls [<*vpls-id*> [<*mac-address*>]]

The <*vpls-id*> variable is the ID of a VPLS instance. If you specify the VPLS ID, you can also specify a particular entry in the VPLS MAC database by adding the optional <*mac-address*> variable.

Table 257 lists the output displayed by the **show mac vpls** command.

TABLE 257 Output from the **show mac vpls** command

Field	Description
Total VPLS mac entries in the table	The number of MAC addresses that have been learned in the database.
Local	The number of locally learned entries in the database.
Remote	The number of remotely learned entries in the database.
VPLS	The VC ID of the VPLS instance.
MAC Address	The MAC address of the entry.
L/R	Whether the entry was learned from local endpoints (L), or was learned from a remote VPLS peer (R).
Port	The port number for the entry.
Vlan:Inner-VLAN/Peer	For Local entries, the VLAN ID for the port; for dual-tagged VLANs, the outer VLAN ID followed by the inner VLAN ID; for Remote entries, the IP address of the VPLS peer.
Age	The age of the entry. The value on the MP is zero because the aging occurs on line card processors.

NOTE
The information displayed in the SA-CAM and DA-CAM index fields is not relevant for day-to-day management of the device. The information is used by engineering and technical support staff for debugging purposes.

Displaying VPLS traffic statistics

You can display VPLS traffic statistics, to view the forwarding counters for each VPLS configured on the system. The output shows a given port range that receives traffic, how many packets are sent out on local CE device endpoints, and how many are sent out of LSP tunnels to remote PE devices. If the port is a 10G port, a single port is displayed. If the module is a 40x1G module, a range of 10 1G ports is displayed.

NOTE
When CPU protection is on, flooded traffic received from an endpoint is not accounted by the VPLS statistics for endpoint-out packets even though they are locally switched.

To display all VPLS traffic statistics on a router, enter the following command.

```
NetIron# show mpls statistics vpls
VPLS-Name      In-Port(s)     Endpt-Out-Pkts    Tnl-Out-Pkts
---------      ----------     --------------    ------------
test2          e1/1           0                 0
               e1/2           0                 0
               e1/3           0                 0
               e1/4           0                 0
test2          e2/1 - e2/10   0                 0
               e2/11 - e2/20  0                 0
               e2/21 - e2/30  0                 0
               e2/31 - e2/40  0                 0
test3          e1/1           0                 0
               e1/2           0                 0
               e1/3           0                 0
               e1/4           0                 0
test3          e2/1 - e2/10   0                 0
               e2/11 - e2/20  0                 0
               e2/21 - e2/30  0                 0
               e2/31 - e2/40  0                 0
test4          e1/1           0                 0
               e1/2           0                 0
               e1/3           0                 0
               e1/4           0                 0
test4          e2/1 - e2/10   0                 0
               e2/11 - e2/20  0                 0
               e2/21 - e2/30  0                 0
               e2/31 - e2/40  0                 0
test4          e5/1           10354120822       0
               e5/2           0                 0
               e5/3           0                 2992416134
               e5/4           0                 0
```

NOTE
The VPLS name is repeated for each module from which the statistics are collected, to be displayed on the MP console.

To display VPLS traffic statistics for a VPLS instance specified by its VPLS name, enter the following command.

```
NetIron# show mpls statistics vpls test4
VPLS-Name      In-Port(s)     Endpt-Out-Pkts    Tnl-Out-Pkts
---------      ----------     --------------    ------------
test4          e1/1           0                 0
               e1/2           0                 0
               e1/3           0                 0
               e1/4           0                 0
test4          e2/1 - e2/10   0                 0
               e2/11 - e2/20  0                 0
               e2/21 - e2/30  0                 0
               e2/31 - e2/40  0                 0
test4          e5/1           10828448712       0
               e5/2           0                 0
               e5/3           0                 3025869251
               e5/4           0                 0
```

To display VPLS traffic statistics for a VPLS instance specified by its VPLS ID, enter the following command.

```
NetIron# show mpls statistics vpls 4
VPLS-Name      In-Port(s)     Endpt-Out-Pkts    Tnl-Out-Pkts
---------      ----------     --------------    ------------
test4          e1/1           0                 0
               e1/2           0                 0
               e1/3           0                 0
               e1/4           0                 0
test4          e2/1 - e2/10   0                 0
               e2/11 - e2/20  0                 0
               e2/21 - e2/30  0                 0
               e2/31 - e2/40  0                 0
test4          e5/1           10828448712       0
               e5/2           0                 0
               e5/3           0                 3025869251
               e5/4           0                 0
```

Syntax: show mpls statistics vpls [<*vpls-name*> | <*vpls-id*>]

The <*vpls-name*> variable is the configured name for a VPLS instance.

The <*vpls-id*> variable is the ID of a VPLS instance.

Table 258 lists the output displayed by the **show mpls statistics vpls** command.

TABLE 258 Output from the **show mpls statistics vpls** command

Field	Description
VPLS-Name	The configured name of the VPLS instance.
In-Port(s)	The port where the traffic is received.
Endpt-Out-Pkts	The number of packets transmitted out of local endpoints.
Tnl-Out-Pkts	The number of packets transmitted out of LSP tunnels.

Clearing VPLS traffic statistics

To clear the entries stored for all VPLS statistics, enter the following command.

```
NetIron# clear mpls statistics vpls
```

Syntax: clear mpls statistics vpls [<*vpls-name*> | <*vpls-id*>]

The <*vpls-name*> variable is the configured name for a VPLS instance.

The <*vpls-id*> variable is the ID of a VPLS instance.

The support enables simplified interactions between MPLS and VPLS with regard to VPLS peer FSM transitions. The LDP integration is supported on all platforms.

VPLS LDP

Displaying the VPLS peer FSM state with LDP support

You can display the various VPLS peer FSM states with the LDP integration on the device using the **show mpls vpls** commands.

Table 259 provides a description of all the peer FSM states with the LDP support.

TABLE 259 PEER FSM state description

Peer FSM state name	state description
Wait for functional local ports	No functional local endpoints.
Wait for LSP tunnel to Peer	No LSP tunnels available to reach the remote peer.
Wait for PW UP (Wait for LDP Session)	LDP session to remote peer is down.
Wait for PW UP (Wait for remote VC label)	PW is down (waiting for remote peer's VC label.)
Wait for PW UP (VC type Mismatched)	PW is down (VC type mismatched).
Wait for PW UP (MTU Mismatched)	PW is down (MTU mismatched).
Wait for PW UP (VC Bind In Progress)	PW is down (Local VC binding in progress).
Operational	PW is up and operational.
Wait Withdraw Done ...	Waiting for VC withdraw completion (internal intermediate states).
VC BIND Failure State	VC binding failed. User intervention required.
VC Withdraw Failure State	VC withdraw failed. User intervention required.

User intervention is required to recover from the VC Bind Failure state, and the VC Withdraw Failure state. To recover, you must delete the failed peer and then add it back. These failure states may occur during extreme conditions when the system runs out of memory to issue ITC requests. When these failures are detected, VPLS generates the following syslog messages accordingly.

```
WARN: VPLS id X Peer IP Address: aa.bb.cc is placed in VC Bind Failure state due
to low system memory.
WARN: VPLS id Y Peer IP Address: dd.ee.ff  is placed in VC Withdraw Failure state
due to low system memory.
```

VC type mismatched

The following example shows the output for the LDP integration for a VC type mismatched case.

```
NetIron# show mpls vpls id 200
VPLS vc_mismatched, Id 200, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4098
  Vlan 200 inner-vlan 145
   Tagged: ethe 2/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 33.33.33.33, State: Wait for PW Up (VC type mismatched)
  Tnnl in use: tnl0(2)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: N/A
  LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: Ethernet (0x05)
 CPU-Protection: OFF
 Local Switching: Enabled
```

The local VC label and remote VC label display will be performed only if the Peer is in Operational state. Else, it will display N/A for these fields.

The remote VC Type will be the same as the local VC type if the peer state is Operational, else, it will be shown as N/A.

MTU mismatched

The following example shows the output for the LDP integration for a MTU mismatched case.

```
NetIron# show mpls vpls id 300
VPLS mtu_mismatched, Id 300, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4099
  Vlan 100 inner-vlan 145
   Tagged: ethe 1/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 44.44.44.44, State: Wait for PW Up (MTU mismatched)
  Tnnl in use: tnl3(3)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 2500,
  LOCAL VC-Type: Ethernet Tagged (0x04), Ethernet Tagged (0x04)
 CPU-Protection: OFF
Local Switching: Enabled
```

No remote VC label

The following example shows the output for the LDP integration for a no remote VC label case.

```
NetIron# show mpls vpls id 400
VPLS waiting_for_remote_label, Id 400, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4100
  Vlan 900 inner-vlan 245
   Tagged: ethe 7/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 55.55.55.55, State: Wait for PW Up (Wait for remote VC label)
  Tnnl in use: tnl4(5)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0,
  LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
 CPU-Protection: OFF
 Local Switching: Enabled
```

LDP session down

The following example shows the output for the LDP integration for an LDP session down case.

```
NetIron# show mpls vpls detail
VPLS NO_LDP, Id 500, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4101
  Vlan 880 inner-vlan 35
   Tagged: ethe 8/2
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 66.66.66.66, State: Wait for PW Up (Wait for LDP session to
Peer)
  Tnnl in use: tnl5(6)
  LDP session: Down, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0,
  LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
 CPU-Protection: OFF
 Local Switching: Enabled
```

No local label resource

The following example shows the output for the LDP integration for a no local label resource case.

```
NetIron# show mpls vpls detail
VPLS waiting_for_remote_label, Id 400, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4100
  Vlan 900 inner-vlan 245
   Tagged: ethe 7/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 55.55.55.55, State: Wait for PW Up (No Label Resource)
  Tnnl in use: tnl4(5)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0,
  LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
 CPU-Protection: OFF
 Local Switching: Enabled
```

MPLS LDP show commands

If there are issues with the peer VC labels (local or remote), MTU values, or VC type, other than using the vpls show command as documented in previous sections, one can also use the **mpls ldp show** command to compare the PW VC information.

Using the show mpls ldp vc x command

Here is an example of the **show mpls ldp fec vc** command where the remote peer is in the Operational state.

```
NetIron#show mpls ldp fec vc 1
FEC_CB: 0x34ff85a8, idx: 50, type: 128, pend_notif: None
 State: current, Ingr: Yes, Egr: Yes, UM Dist. done: Yes
 VC-Id: 1, vc-type: 4, grp-id: 0
 Local-mtu: 2000, remote-mtu: 1500, MTU enforcement: disabled
 Downstream mappings:
 Local LDP ID        Peer LDP ID         Label     State     CB
 21.21.21.21:0       11.11.11.11:0       983041    Installed 0x34eb2140(-1)
 Upstream mappings:
 Local LDP ID        Peer LDP ID         Label     CB
 21.21.21.21:0       11.11.11.11:0       983040    0x34eb2510(-1)
```

Configuring BGP-Based Auto-Discovery for VPLS 34

Overview

The following list displays the BGP-Based Auto-Discovery for VPLS features supported by PowerConnect B-MLXe:

- BGP-Based Auto-Discovery for VPLS

This chapter describes how to configure the PowerConnect device to automatically discover Virtual Private LAN Services (VPLS) endpoints that are part of the same VPLS domain.

VPLS is a method for carrying Layer 2 frames between Customer Edge (CE) devices across a Multi-Protocol Label Switched (MPLS) domain. Information about VPLS, how it works, and how to manually configure it is discussed in the 33, "Configuring MPLS Virtual Private LAN Services".

The implementation of ***BGP-based auto-discovery for VPLS*** (also called ***VPLS auto-discovery***) eliminates the need for manual configuration of VPLS peers for every VPLS instance configured on the device. The implementation complies with the Internet draft *draft-ietf-l2vpn-signaling-08*. Using the services of Border Gateway Protocol version 4 (BGP4) and Label Distribution Protocol (LDP), VPLS auto-discovery enables a device to automatically discover other VPLS Provider Edge (PE) devices that are part of the same VPLS domain, and to detect and converge when other PE routers are added to or removed from the VPLS domain.

Terms introduced in this chapter

BGP-based auto-discovery for VPLS – Also called ***VPLS auto-discovery***, this feature enables automatic discovery of VPLS Provider Edge (PE) devices that are part of the same VPLS domain, and the ability to detect and converge when other PE routers are added to or removed from the VPLS domain.

BGP L2VPN VPLS Routing Information Base (RIB) – Also called the **BGP L2VPN RIB**, this is the database that contains information about VPLS endpoints that are automatically discovered through VPLS auto-discovery.

L2VPN VPLS address family or **L2VPN address family** – This is the BGP-based auto-discovery mechanism used to distribute information about VPLS endpoints. Information is stored in the BGP L2VPN VPLS Routing Information Base.

Label Switch Router (LSR) ID – This is the router ID. LDP assigns the default loopback address as the router ID. Since VPLS auto-discovery uses the services of LDP, a valid loopback address must be configured on the PowerConnect before VPLS auto-discovery can be enabled.

Route Distinguisher (RD) – The address qualifier used within a single Internet Service Provider's (ISP's) Multi-Protocol Label Switching (MPLS) network. The qualifier is used to distinguish the distinct Virtual Private Network (VPN) routes of separate customers who connect to the service provider.

Route Target (RT) Extended Community – Defines the import and export policies applied to a VPLS instance. Each VPLS instance is associated with one or more route target extended communities.

Subsequent Address Family Identifier (SAFI) – An ID number that provides additional information about the NLRI type for a given attribute.

VPLS Virtual Circuit Identifier (VPLS VCID) or ***VPLS ID*** – Identifies the endpoints of a VPLS instance. All Provider Edge (PE) routers that are part of the same VPLS instance have the same VPLS VCID.

How BGP-based auto-discovery for VPLS works

The devices use the services of LDP and BGP4 to automatically discover VPLS endpoints that are part of the same VPLS domain. To enable the PowerConnect to distribute information about VPLS endpoints, you must configure a L2VPN VPLS address family, activate BGP peering on the L2VPN VPLS address family, then enable BGP-based auto-discovery for VPLS. When BGP L2VPN VPLS update messages are exchanged between PE routers, the device can start discovering VPLS peer addresses.

For every VPLS instance on which BGP-based auto-discovery is enabled, the device automatically generates a Route Distinguisher (RD) value based on the BGP Autonomous System (AS) number and the VPLS Virtual Circuit Identifier (VCID) for PE routers. The RD Is an address qualifier used by the PE router to distinguish VPN routes of separate customers. Also, if not manually configured, the device automatically generates import and export route targets that define the policies that each VPLS instance will use. A local VPLS endpoint Network Layer Reachability Information (NLRI) and import route-target tree are also created and sent to BGP peers with the L2VPN VPLS capability.

When the device receives information about a VPLS endpoint, it checks if its extended community matches any locally-configured VPLS import route targets. If a match is found, information about the VPLS endpoint is stored in the BGP L2VPN routing table and a notification is sent to VPLS for peering information. Once VPLS receives the information, it creates a VPLS peer and starts a peering session.

When VPLS auto-discovery is disabled for a VPLS instance, the system removes all auto-discovered peers for the VPLS instance from the configuration. It then removes the route (local VPLS endpoint address) from the BGP L2VPN route table and sends a “withdrawn” message to VPLS peers, prompting them to remove the route and to disable VPLS auto-discovery. Finally, the system updates the route target tree and sends a route refresh message for the L2VPN VPLS address family.

About the L2VPN VPLS address family

The ***L2VPN VPLS address family*** is an integral part of BGP-based auto-discovery for VPLS, in that it is the mechanism used by BGP4 to distribute information about VPLS endpoints. The L2VPN address family is configured at the BGP configuration level of the CLI and supports the VPLS Subsequent Address Family Identifier (SAFI), an address qualifier that provides additional information about the Network Layer Reachability Information (NLRI) type for a given attribute.

BGP4 uses the L2VPN address family to build the BGP L2VPN Routing information Base (RIB). The L2VPN database is updated each time a Layer 2 Virtual Forwarding Instance (VFI) is configured.

Information about configuring the L2VPN Address Family is in the section “Configuring the L2VPN VPLS address family and activating the BGP4 peering session” on page 1551.

Feature limitations and configuration notes

Consider teh following feature limitations and configuration naotes:

- VPLS should not be used if FDP is enabled.
- VPLS auto-discovery and manual configuration of VPLS peers are supported together on the same device. However they are not supported together on the same VPLS instance.

Scalability

The following section describes the scalability:

- The maximum number of BGP4 peers that can support the L2VPN VPLS address family is equal to the maximum number of BGP4 peers supported on the device.
- The maximum number of VPLS instances that can support BGP-based auto-discovery for VPLS is equal to the maximum number of VPLS instances supported on the device.
- The maximum number of BGP-based auto-discovered peers supported per VPLS instance is equal to the maximum number of unique VPLS or VLL peers or number of VPLS peers supported on the device. If the system exceeds the default or manually-configured maximum number of VPLS peers supported on the device, any new peering for VPLS auto-discovery is rejected.

NOTE
For more information about setting the maximum number of VPLS instances and peers on the device, refer to "Specifying the maximum number of VPLS instances on the device" on page 1498.

Configuring BGP-based auto-discovery for VPLS

It is recommended that you perform the configuration tasks in the order listed in Table 260. Performing the tasks in the recommended sequence minimizes CPU consumption and route flapping. Except where noted as "optional", the configuration tasks in the table are required for VPLS auto-discovery.

TABLE 260 Configuration tasks for VPLS auto-discovery

Configuration task		See...
1	Configure a loopback address	"Configuring a loopback interface" on page 1544
2	Enable BGP4 and assign a local Autonomous System (AS) number	"Configuring BGP4 to support VPLS auto-discovery" on page 1545
3	Enable MPLS and configure LDP	To configure MPLS, refer to the 30, "Configuring MPLS Traffic Engineering". To configure LDP, refer to the 31, "Configuring Label Distribution Protocol (LDP)".
4	Configure VPLS: • Create a VPLS instance • Define the route target (optional) • Enable load balancing (optional)	"Configuring VPLS to support auto-discovery" on page 1547

TABLE 260 Configuration tasks for VPLS auto-discovery

Configuration task		See...
5	Enable VPLS auto-discovery	"Enabling VPLS auto-discovery" on page 1550
6	Configure the L2VPN VPLS address family and activate BGP4 peering	"Configuring the L2VPN VPLS address family and activating the BGP4 peering session" on page 1551

After performing the configuration steps listed in Table 260, you can observe the L2VPN VPLS address family routes, neighbor summary, and VPLS auto-discovery peering. Refer to "Displaying VPLS auto-discovery information" on page 1556.

Configuring a loopback interface

You must configure a loopback address on the PowerConnect before enabling VPLS auto-discovery.

This section contains the following topics:

- "About loopback interfaces and the router ID"
- "Changes that occur when a loopback interface is deleted"
- "Adding a loopback interface"
- "Viewing the loopback Interface"

About loopback interfaces and the router ID

In most configurations, a PowerConnect has multiple IP addresses, usually configured on different interfaces. As a result, a PowerConnect's identity to other devices varies depending on the interface to which the other device is attached. BGP4 identifies a PowerConnect by just one of the IP addresses configured on the device, regardless of the interfaces that connect the devices. This IP address is the ***router ID*** also known as the Label Switched Router (LSR) ID.

LDP uses the default loopback address as the router ID. Since VPLS auto-discovery uses the services of LDP, a valid loopback address must be configured on the PowerConnect before VPLS auto-discovery can be enabled. If a loopback address is not configured, the LDP router ID will be NULL and VPLS auto-discovery will not function.

If there are several loopback addresses configured on the device, the default loopback address is the IP address configured on the lowest-numbered loopback interface on the PowerConnect. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default router ID is 9.9.9.9/24.

```
Loopback interface 1, 9.9.9.9/24
Loopback interface 2, 4.4.4.4/24
Loopback interface 3, 1.1.1.1/24
```

Changes that occur when a loopback interface is deleted

If a loopback interface is deleted while VPLS auto-discovery is enabled, and more than one loopback interface is configured on the device, the PowerConnect uses the IP address configured on the next lowest numbered loopback interface as the router ID. For example, using the following configuration, if loopback interface 1 is deleted, the PowerConnect uses loopback interface 2. Thus, the successive router ID is 4.4.4.4/24. The system will remove all existing VPLS routes for 9.9.9.9/24 and obtain new routes for 4.4.4.4/24.

```
Loopback interface 1, 9.9.9.9/24
```

```
Loopback interface 2, 4.4.4.4/24
```

If a loopback interface is deleted from the configuration while VPLS auto-discovery is enabled, and there are no other valid loopback interfaces, the system will disable LDP and VPLS auto-discovery.

Adding a loopback interface

To add a loopback interface, enter commands such as the following.

```
NetIron(config)# int loopback 1
NetIron(config-lbif-1)# ip address 10.1.1.4/24
```

Syntax: [no] interface loopback <*num*>

Use the **no** form of the command to delete a loopback interface. Also refer to "Changes that occur when a loopback interface is deleted"in the following section.

The <*num*> value can be a number from 1 – 64.

Viewing the loopback Interface

Use the **show mpls ldp** command to view the loopback interface and router ID in use on the router. Refer to "Displaying information about LDP" on page 1573.

Configuring BGP4 to support VPLS auto-discovery

BGP4 must be enabled on the device and a local ***Autonomous System*** (AS) number must be assigned before VPLS auto-discovery can be enabled.

This section includes configuration details for the following BGP-related tasks:

- How to enable BGP4 and assign the local AS number
- How to change or clear the local AS number when VPLS auto-discovery is enabled
- How to disable BGP4 when VPLS auto-discovery is enabled

NOTE
This section provides minimal information about configuring BGP4 neighbors, peer groups, and other essential BGP-related configuration tasks, because its focus is to provide information about configuring BGP to support VPLS auto-discovery. For more information about essential configuration tasks for BGP4, refer to 26, "Configuring BGP4 (IPv4)".

Enabling BGP4 and assigning the local AS number

In a VPLS configuration, all PEs in the same VPLS domain must be configured with the same AS number. To assign a local AS number to the PowerConnect device, enter commands such as the following.

```
NetIron(config)#router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
NetIron(config-bgp)#local-as 10
NetIron(config-bgp)#neighbor 10.1.1.1 remote-as 10
```

These commands enable BGP4, set the local AS number to 10, and add the IP address of the remote neighbor (10.1.1.1) to the IPv4 multi-protocol BGP neighbor table of the local router.

Syntax: [no]router bgp

Syntax: [no] local-as <*number*>

Syntax: [no] neighbor <*IPv4-address*> **remote-as** <*number*>

For local-as <*number*>, specify the AS number in which the router you are configuring resides.

For <*IPv4-address*>, enter the IP address of the remote neighbor.

For remote-as <*num*>, enter the AS number.

Changing or clearing the local AS number when VPLS auto-discovery is enabled

If VPLS auto-discovery is enabled on the device and you want to clear or change the BGP local AS number, you must first disable VPLS auto-discovery, then clear the local AS number. If you attempt to clear or change the local AS number while VPLS auto-discovery is enabled, the console will display the following message.

```
NetIron(config-bgp)#no router bgp
Error:  VPLS instances with BGP auto-discovery exist, remove auto-discovery
configuration first!
```

To clear the BGP local AS number when VPLS auto-discovery is enabled, enter commands such as the following.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls c1 10
NetIron(config-mpls-vpls-c1)#no auto-discovery
NetIron(config-mpls-vpls-c1)#router bgp
NetIron(config-bgp)#no local-as 10
BGP is no longer operational
```

Syntax: [no] auto-discovery

Syntax: [no] local-as <*num*>

Disabling BGP4 when VPLS auto-discovery is enabled

If VPLS auto-discovery is enabled on the device and you want to disable BGP4, first disable VPLS auto-discovery, then disable BGP4 at the VPLS instance level of the CLI. If you attempt to disable BGP4 while VPLS auto-discovery is enabled, the console will display the following message.

```
NetIron(config-bgp)#no router bgp
Error:  There are VPLS instances with BGP auto-discovery enabled, disable
auto-discovery first!
```

NOTE
If VPLS auto-discovery is not enabled on the device, you can disable BGP simply by entering the CLI command **no router bgp** at the global CONFIG level of the CLI.

To disable BGP4 when VPLS auto-discovery is enabled, enter commands such as the following.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls c1 10
NetIron(config-mpls-vpls-c1)#no auto-discovery
NetIron(config-mpls-vpls-c1)#no router bgp
router bgp mode now disabled.  All bgp config data will be lost when writing to
flash!!
```

Syntax: [no] auto-discovery

Syntax: [no] router bgp

NOTE
When BGP is disabled, the system also removes the BGP local AS number from the configuration.

Configuring VPLS to support auto-discovery

This section describes how to configure VPLS to support BGP-based auto-discovery. It includes the following configuration details:

- "Creating a VPLS instance"
- "Defining the route target for a VPLS instance (optional)"
- "Enabling and disabling load balancing for a VPLS instance (optional)"

NOTE
This section provides minimal information about configuring VPLS and other related configuration tasks, because its focus is to provide information about configuring VPLS to support BGP-based auto-discovery. For more information about essential configuration tasks for VPLS, refer to 33, "Configuring MPLS Virtual Private LAN Services".

Creating a VPLS instance

To create a VPLS instance, enter VPLS configuration statements on two or more PE routers.

On the PE routers, enter commands such as the following.

```
NetIron(config)# router mpls
NetIron(config-mpls)#mpls-interface ethernet 1/1
NetIron(config-mpls)# vpls CustomerA 10
NetIron(config-mpls-vpls-CustomerA)#
```

On the VPLS peers (if they are Dell devices), enter commands similar to the above.

```
NetIron(config)# router mpls
NetIron(config-mpls)#mpls-interface ethernet 6/1
NetIron(config-mpls)# vpls CustomerA 10
NetIron(config-mpls-vpls-CustomerA)#
In the above configurations, the endpoints of the VPLS instance are associated by
having the same Virtual Circuit Identifier (VCID) of 10 on each PE router.
```

Syntax: [no] router mpls

Syntax: [no]mpls-interface ethernet [<*slot-num*>/]<*portnum*>

Syntax: [no] vpls <*name*> <*vpls-vcid*>

The **router mpls** command enables MPLS. Enter the **no** form of the command to disable it.

The **mpls-interface ethernet** command specifies the interface on which to create the VPLS instance.

The **vpls** <*name*> parameter specifies the VPLS instance name. The name can be up to 64 alphanumeric characters.

The <*vpls-vcid*> parameter specifies the VCID for the BGP L2VPNVPLS instance. The endpoints of a VPLS instance are associated by having the same VCID on each PE router. Enter a number in the range 1 – 4294967294.

Defining the route target for a VPLS instance (optional)

NOTE
If you opt to manually define a route target, it is recommended that you do so before enabling VPLS auto-discovery.

The ***route target*** extended community for VPLS auto-discovery defines the import and export policies that a VPLS instance will use. The export route target sets an extended community attribute number that is appended to all routes that are exported from the VPLS instance. The import route target value sets a filter that determines the routes that will be accepted into the VPLS instance. Any route with a value in its import route target contained in its extended attributes field matching the value in the VPLS instance's import route target will be accepted. Otherwise the route will be rejected.

In a configuration with VPLS auto-discovery, configuring a route target is optional. If you do not manually configure one, the system will automatically generate the import and export route target for each VPLS instance configured on the PowerConnect device when VPLS auto-discovery is enabled. A manually-configured route target takes precedence over one that is automatically generated by VPLS auto-discovery. If all manually-configured route targets are removed from a VPLS instance while VPLS auto-discovery is enabled, the system will automatically generate a new route target for the VPLS instance.

The PowerConnect supports up to 16 unique import and export route targets per VPLS instance. If you attempt to configure more than 16, the system will display the following error message.

```
Error: Maximum number of Import RT for a VPLS instance is 16!
Error: Maximum number of Export RT for a VPLS instance is 16!
```

To define an import route target of 3:6 and an export route target of 3:8 for a VPLS instance, enter the following commands.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls c1
NetIron(config-mpls-vpls-c1)#route-target import 3:6
NetIron(config-mpls-vpls-c1)#route-target export 3:8
```

Syntax: [no] route-target [both | import | export] <*ASN:num*> **|** <*IP-address:num*>

The **both** parameter specifies both import and export values will apply to the specified route target for the VPLS instance where this command is applied. This is the default state and will apply if no specific value for this parameter is set.

The **import** parameter specifies that routes with route-target extended community attributes matching the specified route-target can be imported into the VPLS instance where this command is applied.

The **export** parameter specifies the route-target extended community attributes that are attached to routes exported from the specified VPLS instance.

The <*ASN:num*> parameter identifies the route as an ASN relative. This number is the local ASN number followed by a colon (:) and a unique arbitrary number.

The <*IP-address:num*> parameter identifies the route as an IP-address relative. This number is the local IP address followed by a colon (:) and a unique arbitrary number.

Viewing the route target for a VPLS instance

Use the **show mpls vpls name** command to view the route targets for a VPLS instance. Refer to "Displaying information about VPLS auto-discovery and load balancing" on page 1571.

Enabling and disabling load balancing for a VPLS instance (optional)

This section describes how to enable and disable load balancing for a VPLS instance on which VPLS auto-discovery is enabled. When load balancing is enabled, the PowerConnect will automatically load balance traffic to all auto-discovered peers.

You can configure a VPLS instance to load balance known unicast traffic sent to auto-discovered VPLS peers across multiple tunnel LSPs. The CLI commands for enabling and disabling load balancing differ depending on whether VPLS auto-discovery is enabled on the VPLS instance. Follow the appropriate procedures in this section.

NOTE
To enable load balancing on VPLS peers that are manually created, refer to "LSP load balancing for VPLS traffic" on page 1504.

NOTE
The PowerConnect load balances traffic for auto-discovered VPLS peers, the same as for manually-created VPLS peers. For details about how traffic is load balanced from one VPLS peer to another, refer to "LSP load balancing for VPLS traffic" on page 1504.

Enabling load balancing when VPLS auto-discovery is disabled

To enable VPLS auto-discovery and load balancing of traffic sent to auto-discovered VPLS peers, enter commands such as the following:

NOTE
Before enabling VPLS auto-discovery, make sure you have completed the configuration tasks listed in Table 260 on page 1543.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls c1 10
NetIron(config-mpls-vpls-c1)#auto-discovery load-balance
```

Syntax: [no] auto-discovery load-balance

To disable load balancing, refer to "Disabling load balancing".

Enabling load balancing when VPLS auto-discovery is enabled

If VPLS auto-discovery is enabled for a VPLS instance and you wish to enable load balancing, you must first disable VPLS auto-discovery, then re-enable it with the **load-balancing** option. If you attempt to enable load balancing when VPLS auto-discovery is enabled, the console will display the following message.

```
NetIron(config-mpls-vpls-c1)#auto-discovery load-balance
Error:  Please disable auto-discovery before make change!
```

To enable load balancing for a VPLS instance that has VPLS auto-discovery enabled, enter commands such as the following.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls c1 10
NetIron(config-mpls-vpls-c1)#no auto-discovery
NetIron(config-mpls-vpls-c1)#auto-discovery load-balance
```

The above commands disable VPLS auto-discovery for VPLS instance "c1", then re-enable VPLS auto-discovery with the **load-balance** option.

Syntax: [no] auto-discovery

Syntax: [no]auto-discovery load-balance

Disabling load balancing

To disable load balancing when VPLS auto-discovery is enabled on the device, first disable VPLS auto-discovery, then re-enable it without the **load-balancing** option.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls c1 10
NetIron(config-mpls-vpls-c1)#no auto-discovery load-balance
NetIron(config-mpls-vpls-c1)#auto-discovery
```

Syntax: [no]auto discovery load-balance

Syntax: [no]auto-discovery

Viewing the load balancing configuration

Use the **show mpls vpls name** command to view If VPLS traffic to the peer is load balanced across tunnel LSPs, and the tunnel LSPs used to reach the peer. Refer to "Displaying information about VPLS auto-discovery and load balancing" on page 1571

Enabling VPLS auto-discovery

NOTE
Before enabling VPLS auto-discovery, make sure you have completed the configuration tasks listed in Table 260 on page 1543.

To enable auto-discovery for a VPLS instance, enter commands such as the following.

```
NetIron(config)#router mpls
NetIron(config-mpls)#vpls c1
NetIron(config-mpls-vpls-c1)#auto-discovery
```

These commands enable MPLS, then change the CLI configuration level from the global MPLS level to the configuration level for the VPLS instance "c1". The **auto-discovery** command enables auto-discovery for this VPLS instance.

Syntax: [no] auto-discovery

Use the **no** form of the command to disable VPLS auto-discovery.

Configuration notes

Consider the following configuration notes while enabling VPLS auto-discovery:

- If you attempt to enable VPLS auto-discovery without first adding a loopback interface, the following error message will display on the console.

  ```
  NetIron(config-mpls-vpls-c2)auto-discovery
  Error: Please configure a loopback address for LDP first!
  ```

  To add a loopback interface, follow the configuration instructions in "Configuring a loopback interface" on page 1544.

- If you attempt to enable VPLS auto-discovery without first configuring the BGP AS number, the following error message will display on the console.

```
NetIron(config-mpls-vpls-c2)auto-discovery
Error: Cannot configure auto-discovery before configuring BGP-AS number!
```

To configure the BGP AS number, follow the configuration instructions in "Configuring BGP4 to support VPLS auto-discovery" on page 1545.

Configuring the L2VPN VPLS address family and activating the BGP4 peering session

This section describes how to configure the L2VPN VPLS address family and activate BGP4 peering. More information about the L2VPN VPLS address family is in the section "About the L2VPN VPLS address family" on page 1542

NOTE
It is recommended that you activate peering on the L2VPN VPLS address family after performing steps 1 – 5 in Table 260. Otherwise, you will need to clear the entire peering session.

To configure the L2VPN VPLS address family, enter commands such as the following.

```
NetIron(config)#router bgp
NetIron(config-bgp)#address-family l2vpn vpls
NetIron(config-bgp-l2vpn-vpls)#neighbor 10.10.1.1 activate
NetIron(config-bgp-l2vpn-vpls)#exit
```

Syntax: **[no]address-family l2vpn vpls**

Syntax: **[no]neighbor** *<IPv4-address>* **|** *<peer group name>* **activate | remote-as | send-community extended**

The **activate** command enables the exchange and updating of routes within the L2VPN VPLS address family.

The **send-community extended** command enables the sending of extended community attributes to this neighbor.

Clearing the BGP L2VPN route table

You can clear routes from the BGP L2VPN route table with or without resetting the BGP session. Use the appropriate commands in this section.

Clearing the BGP L2VPN route table and resetting BGP

NOTE
This section describes how to clear routes from the BGP L2VPN route table and reset the BGP session. If you do not want to reset the BGP session while clearing routes, refer to "Clearing the BGP L2VPN route table without resetting the BGP session" on page 1552.

You can clear routes from the BGP L2VPN route table that were exchanged by the PowerConnect and:

- All BGP4 neighbors

- A specific neighbor
- A specific peer group

To clear and reset all BGP4 routes from the BGP L2VPN route table, enter the following command.

```
NetIron# clear ip bgp l2vpn vpls neighbor all
```

To clear and reset BGP4 routes exchanged by the PowerConnect and a *specific neighbor*, enter a command such as the following.

```
NetIron# clear ip bgp l2vpn vpls neighbor 10.10.10.1
```

To clear and reset BGP4 routes exchanged by the PowerConnect and a *specific peer group*, enter a command such as the following.

```
NetIron# clear ip bgp l2vpn vpls neighbor peergroup1
```

Syntax: clear ip bgp neighbor all | <*ip-addr*> | <*peer-group-name*>

The all | <*ip-addr*> | <*peer-group-name*> | <*as-num*> specifies the neighbor.

The <*ip-addr*> parameter specifies a neighbor by its IP interface with the PowerConnect.

The <*peer-group-name*> specifies all neighbors in a specific peer group.

Clearing the BGP L2VPN route table without resetting the BGP session

When clearing all BGP4 routes from the BGP L2VPN route table, you can place policy changes into effect without resetting the BGP session. To do so, enter a command such as the following.

```
NetIron(config-bgp)# clear ip bgp l2vpn vpls neighbor all soft in
```

This command updates the inbound routes in the BGP L2VPN route table by comparing the route policies against the route updates that the PowerConnect has stored. The command does not request additional updates from the neighbor or otherwise affect the session with the neighbor.

Syntax: clear ip bgp l2vpn vpls neighbor all | <*ip-addr*> | <*peer-group-name*> soft [in | out]

The **soft** parameter performs a soft reset of the neighbor session, which does not affect the session with the neighbor.

The **in** parameter updates inbound routes.

The **out** parameter updates outbound routes.

NOTE
If you do not specify “in”, the command applies to both inbound and outbound updates.

Example configuration

The following shows a typical VPLS auto-discovery configuration.

PowerConnect1 configuration

The following commands are entered on PowerConnect1.

```
NetIron(config)# int loopback 1
NetIron(config-lbif-1)# ip address 10.1.1.1/24
NetIron(config-lbif-1)#exit
NetIron(config)#router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
NetIron(config-bgp)#local-as 10
NetIron(config-bgp)#neighbor 10.1.1.2 remote-as 10
NetIron(config-bgp)#exit
NetIron(config)# router mpls
NetIron(config-mpls)#mpls-interface ethernet 1/1
NetIron(config-mpls)# vpls C1 10
NetIron(config-mpls-vpls-C1)#auto-discovery
NetIron(config-mpls)#exit
NetIron(config-mpls)# vpls C2 20
NetIron(config-mpls-vpls-C2)#auto-discovery
NetIron(config-mpls-vpls-C2)#exit
NetIron(config-mpls)#exit
NetIron(config)#router bgp
NetIron(config-bgp)#address-family l2vpn vpls
NetIron(config-bgp-l2vpn-vpls)#neighbor 10.1.1.2 activate
NetIron(config-bgp-l2vpn-vpls)#exit-address-family
NetIron(config-bgp)#exit
NetIron(config)#
```

PowerConnect2 configuration

The following commands are entered on PowerConnect2, a peer of PowerConnect1.

```
NetIron(config)# int loopback 1
NetIron(config-lbif-1)# ip address 10.1.1.2/24
NetIron(config-lbif-1)#exit
NetIron(config)#router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
NetIron(config-bgp)#local-as 10
NetIron(config-bgp)#neighbor 10.1.1.1 remote-as 10
NetIron(config-bgp)#exit
NetIron(config)# router mpls
NetIron(config-mpls)#mpls-interface ethernet 1/1
NetIron(config-mpls)# vpls C1 10
NetIron(config-mpls-vpls-C1)#auto-discovery
NetIron(config-mpls)#exit
NetIron(config-mpls)# vpls C2 20
NetIron(config-mpls-vpls-C2)#auto-discovery
NetIron(config-mpls-vpls-C2)#exit
NetIron(config-mpls)#exit
NetIron(config)#router bgp
NetIron(config-bgp)#address-family l2vpn vpls
NetIron(config-bgp-l2vpn-vpls)#neighbor 10.1.1.1 activate
NetIron(config-bgp-l2vpn-vpls)#exit-address-family
NetIron(config-bgp)#exit
NetIron(config)#
```

After applying the above commands, you can use various **show** commands to display information about the VPLS auto-discovery configuration. In the **show** command examples that follow, the lines in bold type indicate the information specific to the VPLS auto-discovery configuration.

NOTE
The **show mpls vpls name** command has changed. The total VC labels allocated field is no longer displayed in the output of the **show mpls vpls name** command.

For field definitions of the **show ip bgp neighbor** command output, refer to "Displaying summary neighbor information" on page 1107. For field definitions of the **show ip bgp l2vpn vpls** command output, refer to "Displaying VPLS auto-discovery information" on page 1556.

```
NetIron1#show ip bgp nei 10.1.1.2
1   IP Address: 10.1.1.2, AS: 10 (IBGP), RouterID: 2.2.2.2, VRF: default-vrf
    State: ESTABLISHED, Time: 0h1m5s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 34 seconds, HoldTimer Expire in 175 seconds
    Minimal Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       1       2         0            0
       Received: 1       4       2         0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: 0h1m5s     ---
    Last Connection Reset Reason: Hold Timer Expired
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  unicast  capability
      Peer Negotiated VPNv4 unicast  capability
      Peer Negotiated L2VPN VPLS address family
      Peer configured for IPV4 unicast  Routes
      Peer configured for VPNv4 unicast  Routes
      Peer configured for L2VPN VPLS address family
    Neighbor Capability Negotiation:
    As-path attribute count: 3

NetIron1#show ip bgp l2vpn vpls sum
  BGP4 Summary
  Router ID: 1.1.1.1   Local AS Number: 10
  Confederation Identifier: not configured
  Confederation Peers:
  Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
  Number of Neighbors Configured: 1, UP: 1
  Number of Routes Installed: 4, Uses 344 bytes
  Number of Routes Advertising to All Neighbors: 2, Uses 88 bytes
  Number of Attribute Entries Installed: 4, Uses 376 bytes
  Neighbor Address  AS#         State   Time     Rt:Accepted Filtered Sent     ToSend
  10.1.1.2          10          ESTAB   0h 7m21s    2        0        2        0

NetIron1#show ip bgp l2vpn vpls
Total number of BGP L2VPN VPLS Routes: 4
Status codes: s suppressed, d damped, h history, * valid, > best, i  internal, S
stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network         Next Hop    Metric LocPrf Weight Path
Route Distinguisher: 10:10
*>  1.1.1.1/32      0.0.0.0     0      100    65535  i<< local VPLS endpoint for C1
*i  2.2.2.2/32      0.0.0.0     0      100    0      i<<remote VPLS endpoint for C1
Route Distinguisher: 10:20
*>  1.1.1.1/32    0.0.0.0     0     100    65535 i
*i  2.2.2.2/32    0.0.0.0     0     100    0     i
```

```
NetIron1#show mpls vpls name c1
VPLS c1, Id 10, Max mac entries: 8192
 Total vlans: 0, Tagged ports: 0 (0 Up), Untagged ports 0 (0 Up)
 Total VPLS peers: 1 (0 Operational)
 auto-discovery enabled, RD 10:10
 export RT   10:10
 import RT   10:10
 Peer address: 2.2.2.2 (auto-discovered), State: Wait for functional local ports
  Tnnl in use: None
  LDP session: Up, Local VC lbl: 983040, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0
 CPU-Protection: OFF
 Local Switching: Enabled


NetIron1#show mpls vpls name c2
VPLS c2, Id 20, Max mac entries: 8192
 Total vlans: 0, Tagged ports: 0 (0 Up), Untagged ports 0 (0 Up)
 Total VPLS peers: 1 (0 Operational)
 auto-discovery enabled, RD 10:20
 export RT   10:20
 import RT   10:20
 Peer address: 2.2.2.2 (auto-discovered), State: Wait for functional local ports
  Tnnl in use: None
  LDP session: Up, Local VC lbl: 983072, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0
 CPU-Protection: OFF
 Local Switching: Enabled
```

Displaying VPLS auto-discovery information

You can display the following information about the VPLS auto-discovery configuration:

- L2VPN VPLS address family and associated routes
- VPLS auto-discovered peers
- Load balancing status for VPLS auto-discovered peers
- LDP configuration, including the loopback interface and router ID

Displaying information about BGP L2VPN VPLS routes

You can use the **show ip bgp l2vpn vpls** command with the parameters listed in Table 261 to view information related to BGP L2VPN VPLS routes.

TABLE 261 Parameters for CLI command show ip bgp l2vpn vpls

Parameter	Displays...	For details, see...
<A.B.C.D or A.B.C.D/L> (route IP address)	The BGP L2VPN VPLS routes for a particular IP route address	page 1559
attribute-entries	AS-path attribute entries	page 1560

TABLE 261 Parameters for CLI command show ip bgp l2vpn vpls

Parameter	Displays...	For details, see...
neighbors	Details about TCP and BGP neighbor connections	page 1562
rd	Details about the route distinguisher	page 1567
routes	Information about BGP L2VPN VPLS routes	page 1568
summary	A summary of the BGP L2VPN VPLS neighbor status	page 1570

Viewing all BGP L2VPN VPLS routes

The **show ip bgp l2vpn vpls** command displays all of the BGP L2VPN VPLS routes. The following shows example output.

```
NetIron1#show ip bgp l2vpn vpls
Total number of BGP L2VPN VPLS Routes: 4
Status codes: s suppressed, d damped, h history, * valid, > best, i  internal, S
stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network          Next Hop    Metric LocPrf Weight Path
Route Distinguisher: 10:10
*>  1.1.1.1/32      0.0.0.0     0    100   65535  i
*i  2.2.2.2/32      0.0.0.0     0    100   0      i

Route Distinguisher: 10:20
*>  1.1.1.1/32      0.0.0.0     0    100   65535  i
*i  2.2.2.2/32      0.0.0.0     0    100   0      i
```

Syntax: show ip bgp l2vpn vpls

Table 262 defines the fields shown in the above example output.

TABLE 262 Output for the show ip bgp l2vpn vpls command

This field...	Displays
Total number of BGP L2VPN VPLS Routes	The number of BGP4 routes in the BGP L2VPN VPLS route table.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route: • s (suppressed) – This route was suppressed during aggregation and thus is not advertised to neighbors. • d (damped) – This route has been dampened (by the route dampening feature), and is currently unusable. • h (history) – Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • * (valid) – The next-hop of this route can be resolved by the routing table. • > (best) – BGP4 has determined that this is the optimal route to the destination. • i (internal) – The route was learned through BGP4. • S (stale) – This route is stale and will be cleaned up.
Origin codes	A list of the characters the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin code can be one of the following: • i - IGP – The routes with this set of attributes came to BGP4+ through IGP • e - EGP – The routes with this set of attributes came to BGP4+ through EGP. • ? - incomplete – The routes came from an origin other than IGP or EGP. For example, they may have been redistributed from OSPF or RIP. **NOTE:** When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Route Distinguisher	A unique ID that is prepended on any address being routed or advertised from a Virtual Routing and Forwarding (VRF) instance. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" (colon) and a unique arbitrary number. For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" (colon) and a unique arbitrary number.
Network	The IP address and network mask of the destination network of the route.
Next Hop	The IP address of the next-hop router.
Metric	The cost of the routes that have this set of attributes.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.

TABLE 262 Output for the show ip bgp l2vpn vpls command (Continued)

This field...	Displays
Weight	The value that this route associates with routes from a specific neighbor. For example, if the router receives routes to the same destination from two BGP4 neighbors, the router prefers the route from the neighbor with the larger weight
Path	The AS path for the route.

Viewing BGP L2VPN VPLS routes for a particular IP route address

The **show ip bgp l2vpn vpls** <*IP route address*>**>** command displays the BGP L2VPN VPLS routes for a particular IP route address.

```
mu2(config-lbif-1)#show ip bgp l2vpn vpls 10.1.1.1
Total number of BGP L2VPN VPLS Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network           Next Hop        Metric LocPrf Weight Path
Route Distinguisher: 10:10
*>  3.3.3.3/32        0.0.0.0         0      100    65535  i
```

Syntax: show ip bgp l2vpn vpls <*IP route address*>

Field definitions for the **show ip bgp l2vpn vpls** <*IP route address*> command are the same as for **show ip bgp l2vpn vpls**. Refer to Table 262.

Viewing BGP L2VPN VPLS route attribute entries

Use the **show ip bgp l2vpn vpls attribute-entries** command to view attribute entries for BGP L2VPN VPLS routes.

```
NetIron1#show ip bgp l2vpn vpls attribute-entries
        Total number of BGP Attribute Entries: 4 (2)
1       Next Hop  :0.0.0.0            Metric   :0              Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 10:10
        AS Path   : (length 0)
        Address: 0x1431f5a2  Hash:108 (0x01000000), PeerIdx 0
        Links: 0x00000000, 0x00000000, nlri: 0x143709e4
        Reference Counts: 1:0:0, Magic: 5
2       Next Hop  :0.0.0.0            Metric   :0              Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 10:20
        AS Path   : (length 0)
        Address: 0x1431f608  Hash:620 (0x01000000), PeerIdx 0
        Links: 0x00000000, 0x00000000, nlri: 0x14370a42
        Reference Counts: 1:0:0, Magic: 6
3       Next Hop  :0.0.0.0            Metric   :0              Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 10:10
        AS Path   : (length 0)
           AsPathLen: 0  AsNum: 0, SegmentNum: 0, Neighboring As: 0, Source As 0
        Address: 0x1431f4d6  Hash:108 (0x01000000), PeerIdx 4000
        Links: 0x00000000, 0x00000000, nlri: 0x14370928
        Reference Counts: 1:0:1, Magic: 3
4       Next Hop  :0.0.0.0            Metric   :0              Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 10:20
        AS Path   : (length 0)
           AsPathLen: 0  AsNum: 0, SegmentNum: 0, Neighboring As: 0, Source As 0
        Address: 0x1431f53c  Hash:620 (0x01000000), PeerIdx 4000
        Links: 0x00000000, 0x00000000, nlri: 0x14370986
        Reference Counts: 1:0:1, Magic: 4
```

Syntax: show ip bgp l2vpn vpls attribute-entries

Table 263 defines the fields shown in the above example output.

TABLE 263 Output for the show ip bgp l2vpn vpls attribute-entries command

This field...	Displays
Total number of BGP Attribute Entries	The number of routes contained in this router's BGP L2VPN VPLS route table.
Next Hop	The IP address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.

TABLE 263 Output for the show ip bgp l2vpn vpls attribute-entries command (Continued)

This field...	Displays
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. **NOTE:** When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • AS `Number` shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • `Router-ID` shows the router that originated this aggregator.
Atomic	Indicates whether the network information in this set of attributes has been aggregated *and* this aggregation has resulted in information loss. **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities to which routes with this set of attributes belong.
Extended Community	The extended community attributes.
AS Path	The AS path through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	This field is an internal value used for debugging purposes only.
Links	This field in an internal value used for debugging purposes only.
Reference Counts	This field is an internal value used for debugging purposes only.

Viewing neighbor connections

Use the **show ip bgp l2vpn vpls neighbors** command to view the details of TCP and BGP neighbor connections.

```
NetIron1#show ip bgp l2vpn vpls neighbors
    Total number of BGP Neighbors: 1
1   IP Address: 10.1.1.2, AS: 10 (IBGP), RouterID: 2.2.2.2, VRF: default-vrf
    State: ESTABLISHED, Time: 0h15m47s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 49 seconds, HoldTimer Expire in 148 seconds
    Minimal Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 3       2       19        0            0
       Received: 1       2       18        0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: 0h15m47s   ---           Rx: 0h15m47s   ---
    Last Connection Reset Reason: Hold Timer Expired
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  unicast  capability
      Peer Negotiated L2VPN VPLS address family
      Peer configured for IPV4 unicast  Routes
      Peer configured for L2VPN VPLS address family
    Neighbor AS4 Capability Negotiation:
    As-path attribute count: 2
    TCP Connection state: ESTABLISHED, flags:00000044 (0,0)
     Maximum segment size: 1460
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:   604, Received: 585
       Local host:  10.1.1.1, Local  Port: 179
       Remote host: 10.1.1.2, Remote Port: 8018
       ISentSeq:  310843582  SendNext:  310844187  TotUnAck:          0
       TotSent:         605  ReTrans:           0  UnAckSeq:  310844187
       IRcvSeq:   310909513  RcvNext:   310910099  SendWnd:       64981
       TotalRcv:        586  DupliRcv:          0  RcvWnd:        65000
       SendQue:           0  RcvQue:            0  CngstWnd:       3102
```

Syntax: show ip bgp l2vpn vpls neighbors

TABLE 264 Output for the show ip bgp l2vpn vpls neighbors command

This field...	Displays...
Total Number of BGP Neighbors	The number of BGP neighbors configured.
IP Address	The IP address of the neighbor.
AS	The AS number to which the neighbor belongs.
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP – The neighbor is in another AS. • EBGP_Confed – The neighbor is a member of another sub-AS in the same confederation. • IBGP – The neighbor is in the same AS.
RouterID	The neighbor's router ID.

TABLE 264 Output for the show ip bgp l2vpn vpls neighbors command (Continued)

This field...	Displays...
VRF	• default-vrf – The L2VPN is only applicable to the global default VRF instance.
State	The state of the router's session with the neighbor. The states are from this router's perspective of the session, not the neighbor's perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each router: • IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. • ADMND – The neighbor has been administratively shut down. • CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. **NOTE:** If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT – BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the router receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED – BGP4 is ready to exchange UPDATE messages with the neighbor.
State (cont'd)	Operational States: Additional information regarding the operational states of the BGP states described above may be added as described in the following: • (+) – Indicates that there is more BGP data in the TCP receiver queue. • (-) – indicates that the session has gone down and the software is clearing or removing routes. • (*) – indicates that the inbound or outbound policy is being updated for the peer. • (s) – indicates that the peer has negotiated restart, and the session is in a stale state. • (r) – indicates that the peer is restarting the BGP connection, through restart. • (^) – On the standby MP indicates that the peer is in the ESTABLISHED state and has received restart capability (in the primary MP). • (<) – Indicates that the router is waiting to receive the "End of RIB" message from the peer.
Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this router sends keep alive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the router will wait for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead.
Minimal Route Advertisement Interval	The minimum time elapse between route advertisements to the same neighbor.
RefreshCapability	Indicates whether this PowerConnect has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.

TABLE 264 Output for the show ip bgp l2vpn vpls neighbors command (Continued)

This field...	Displays...
Messages Sent	The number of messages this router has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this router has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	The last time updates were sent and received for the following: • NLRIs • Withdraws

Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following:
	• Reasons described in the BGP specifications:
	  • Message Header Error
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • OPEN Message Error
	  • Unsupported Version Number
	  • Bad Peer AS Number
	  • Bad BGP Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unsupported Capability
	  • UPDATE Message Error
	  • Malformed Attribute List
	  • Unrecognized Well-known Attribute
	  • Missing Well-known Attribute
	  • Attribute Flags Error
	  • Attribute Length Error
	  • Invalid ORIGIN Attribute
	  • Invalid NEXT_HOP Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS_PATH
	  • Hold Timer Expired
	  • Finite State Machine Error
	  • Rcv Notification

TABLE 264 Output for the show ip bgp l2vpn vpls neighbors command (Continued)

This field...	Displays...

Last Connection Reset Reason (cont'd)	• Reasons specific to the implementation:
	  • Reset All Peer Sessions
	  • User Reset Peer Session
	  • Port State Down
	  • Peer Removed
	  • Peer Shutdown
	  • Peer AS Number Change
	  • Peer AS Confederation Change
	  • TCP Connection KeepAlive Timeout
	  • TCP Connection Closed by Remote
	  • TCP Data Stream Error Detected

Notification Sent	If the router sends a NOTIFICATION message to the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.
	• Message Header Error:
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • Unspecified
	• Open Message Error:
	  • Unsupported Version
	  • Bad Peer As
	  • Bad BGP Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unspecified
	• Update Message Error:
	  • Malformed Attribute List
	  • Unrecognized Attribute
	  • Missing Attribute
	  • Attribute Flag Error
	  • Attribute Length Error
	  • Invalid Origin Attribute
	  • Invalid NextHop Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS Path
	  • Unspecified
	• Hold Timer Expired
	• Finite State Machine Error
	• Cease
	• Unspecified

Notification Received	If the router receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages. See above.

TABLE 264 Output for the show ip bgp l2vpn vpls neighbors command (Continued)

This field...	Displays...
Neighbor NLRI Negotiation	The state of the NLRI negotiation with the neighbor. For example: • Peer negotiated IPv4 unicast capability • Peer negotiated L2VPN VPLS address family • Peer configured for IPv4 unicast routes • Peer configured for L2VPN VPLS address family
Neighbor AS4 Capability Negotiation	Whether this neighbor enabled 4 bytes ASN capability,
As-path attribute count	The number of unique path attributes learned from this neighbor.
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Maximum segment size	The TCP maximum segment size.
TTL check	The TCP TTL check.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the PowerConnect.
Local port	The TCP port the PowerConnect is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the PowerConnect.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the PowerConnect that have not been acknowledged by the neighbor.

TABLE 264 Output for the show ip bgp l2vpn vpls neighbors command (Continued)

This field...	Displays...
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the PowerConnect retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Viewing information for a route distinguisher

Use the **show ip bgp l2vpn vpls rd** command to view information for a particular route distinguisher.

```
NetIron#show ip bgp l2vpn vpls rd 10:10
Total number of BGP Routes: 2
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  1.1.1.1/32         0.0.0.0         0      100    65535  i
*i  2.2.2.2/32         0.0.0.0         0      100    0      i
```

Syntax: show ip bgp l2vpn vpls rd

TABLE 265 Output for the show ip bgp l2vpn vpls rd command

This field...	Displays
Total number of BGP Routes	The number of BGP4 routes the PowerConnect has installed in the BGP L2VPN VPLS route table.
Status codes	A list of the characters the display uses to indicate the route's status. Refer to "Status codes" on page 1558.
Origin codes	A list of the characters the display uses to indicate the route's origin. Refer to "Origin codes" on page 1558.
Network	The IP address and network mask of the destination network of the route.
Next Hop	The IP address of the next-hop router.
Metric	The cost of the route.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.

TABLE 265 Output for the show ip bgp l2vpn vpls rd command (Continued)

This field...	Displays
Weight	The value that this route associates with routes from a specific neighbor. For example, if the router receives routes to the same destination from two BGP4 neighbors, the router prefers the route from the neighbor with the larger weight.
Path	The AS path for the route.

Viewing information about BGP L2VPN VPLS routes

Use the **show ip bgp l2vpn vpls routes** command to view information about BGP L2VPN VPLS routes.

```
NetIron1#show ip bgp l2vpn vpls routes
Total number of BGP Routes: 4
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
Route Distinguisher: 10:10
1      1.1.1.1/32         0.0.0.0         0          100        65535  BL
         AS_PATH:
2      2.2.2.2/32         0.0.0.0         0          100        0      I
         AS_PATH:
Route Distinguisher: 10:20
3      1.1.1.1/32         0.0.0.0         0          100        65535  BL
         AS_PATH:
4      2.2.2.2/32         0.0.0.0         0          100        0      I
         AS_PATH:
```

Syntax: **show ip bgp l2vpn vpls routes**

TABLE 266 Output for the show ip bgp l2vpn vpls routes command

This field...	Displays
Total number of BGP Routes	The number of BGP4 routes the PowerConnect has installed in the BGP4 route table.
Status	A list of the characters the display uses to indicate the route's status. The status code appears in the last column of the display, to the right of each route. The route's status can be one or more of the following: • A: AGGREGATE – The route is an aggregate route for multiple networks. • B: BEST – BGP4 has determined that this is the optimal route to the destination. • b NOT-INSTALLED-BEST – The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the PowerConnect received better routes from other sources (such as OSPF, RIP, or static IP routes). • C: CONFED_EBGP – The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D: DAMPED – This route has been dampened (by the route dampening feature), and is currently unusable. • E: EBGP – The route was learned from another AS BGP neighbor. • F: FILTERED – The route was filtered from the BGP route table. • H: HISTORY – Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I: IBGP – The route was learned from the same AS BGP neighbor. • L: LOCAL – The route originated on this PowerConnect. • M: MULTIPATH – BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". • m: NOT-INSTALLED-MULTIPATH – The software was not able to install the route in the IP route table. • S: SUPPRESSED – This route was suppressed during aggregation and thus is not advertised to neighbors. • s: STALE – This is a stale route and will be cleaned up.
Route Distinguisher	A unique ID that is prepended on any address being routed or advertised from a Virtual Routing and Forwarding (VRF) instance. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" (colon) and a unique arbitrary number. For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" (colon) and a unique arbitrary number.
Prefix	The IP address and network mask of the destination network of the route.
AS_PATH	The BGP AS_PATH path attribute.
Next Hop	The IP address of the next-hop router.
Metric	The cost of this route.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.

TABLE 266 Output for the show ip bgp l2vpn vpls routes command (Continued)

This field...	Displays
Weight	The value that this route associates with routes from a specific neighbor. For example, if the router receives routes to the same destination from two BGP4 neighbors, the router prefers the route from the neighbor with the larger weight.
Status	The route's status. Refer to "Status" on page 1569.

Viewing a summary of BGP neighbor status

Use the **show ip bgp l2vpn vpls summary** command to view BGP4 summary information.

```
NetIron1#show ip bgp l2vpn vpls summary
  BGP4 Summary
  Router ID: 1.1.1.1   Local AS Number: 10
  Confederation Identifier: not configured
  Confederation Peers:
  Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
  Number of Neighbors Configured: 1, UP: 1
  Number of Routes Installed: 4, Uses 344 bytes
  Number of Routes Advertising to All Neighbors: 2, Uses 88 bytes
  Number of Attribute Entries Installed: 4, Uses 376 bytes
  Neighbor Address  AS#         State   Time     Rt:Accepted Filtered Sent     ToSend
  10.1.1.2          10          ESTAB   0h 7m21s    2        0        2        0
```

Syntax: show ip bgp l2vpn vpls summary

TABLE 267 Output for the show ip bgp l2vpn vpls summary command

This field...	Displays
Router ID	The PowerConnect's router ID.
Local AS Number	The BGP4 AS number to which the router belongs.
Confederation Identifier	The AS number of the confederation to which the PowerConnect belongs.
Confederation Peers	The numbers of the local ASs contained in the confederation. This list matches the confederation peer list you configure on the PowerConnect.
Maximum Number of IP ECMP Paths Supported for Load Sharing	The maximum number of route paths across which the device can balance traffic to the same destination.
Number of Neighbors Configured	The number of BGP4 neighbors configured on this PowerConnect, and currently in established state.
Number of Routes Installed	The number of BGP4 routes in the router's BGP4 route table and the route or path memory usage.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors and the amount of memory used by these routes.
Number of Attribute Entries Installed	The number of BGP4 route-attribute entries in the router's route-attributes table and the amount of memory used by these entries.
Neighbor Address	The IP addresses of this router's BGP4 neighbors.
AS#	The AS number.
State	Refer to "State" on page 1563.mmd
Time	The time that has passed since the state last changed.

TABLE 267 Output for the show ip bgp l2vpn vpls summary command (Continued)

This field...	Displays
Rt: Accepted	The number of routes received from the neighbor that this router installed in the BGP4 route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this router filtered out some of the routes received in the UPDATE messages.
Filtered	The routes or prefixes that have been filtered out: • If soft reconfiguration is enabled, this field shows how many routes were filtered out (not placed in the BGP4 route table) but retained in memory. • If soft reconfiguration is not enabled, this field shows the number of BGP4 routes that have been filtered out.
Sent	The number of BGP4 routes that the PowerConnect has sent to the neighbor.
ToSend	The number of routes the PowerConnect has queued to send to this neighbor.

Displaying information about VPLS auto-discovery and load balancing

The **show mpls vpls name** command has changed. The total VC labels allocated field is no longer displayed in the output of the **show mpls vpls name** command.

To display detailed information about a specified VPLS name, enter the following command.

```
NetIron#show mpls vpls name c1
VPLS c1, Id 10, Max mac entries: 8192
 Total vlans: 0, Tagged ports: 0 (0 Up), Untagged ports 0 (0 Up)
 Total VPLS peers: 1 (0 Operational)
 auto-discovery enabled, RD 10:10
 export RT   10:10
 import RT   10:10
 Peer address: 2.2.2.2 (auto-discovered), State: Wait for functional local ports
  Tnnl in use: (load balance): None
  LDP session: Up, Local VC lbl: 983040, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0
 CPU-Protection: OFF
 Local Switching: Enabled
```

Syntax: show mpls vpls name <*name*>

TABLE 268 Output for the show mpls vpls name command

This field...	Displays
VPLS	The configured name of the VPLS instance.
Id	The VCID of this VPLS instance.
Max mac entries	The maximum number of MAC address entries that can be learned for this VPLS instance. This is a soft limit only and can be exceeded if there is space available in the VPLS MAC database.
Total VLANs	The number of VLANs that are translated for this VPLS instance.

TABLE 268 Output for the show mpls vpls name command (Continued)

This field...	Displays
Tagged ports	The total number of tagged ports that are associated with VLANs in this VPLS instance, as well as the number of these ports that are up.
Untagged ports	The total number of untagged ports that are associated with VLANs in this VPLS instance, as well as the number of these ports that are up.
Total VPLS peers	The number of VPLS peers this device has for this VPLS instance, as well as the number of these VPLS peers with which this device has an LDP session.
auto-discovery enabled	Indicates that VPLS auto-discovery is enabled for this VPLS instance.
RD	The Route Distinguisher assigned to the VPLS instance.
export RT	The export route for the VPLS instance.
import RT	The import route for the VPLS instance.
Peer address	The IP address of the VPLS peer. When VPLS auto-discovery is enabled for the VPLS instance, "(auto-discovered)" appears after the IP address.
State	The current state of the connection with the VPLS peer. The VC label allocation is now managed by MPLS. This can be one of the following: • **Operational** – The VPLS instance is operational. Packets can flow between the device and the peer. • **Wait for functional local ports** – The physical endpoint port that should be connected to the Customer Edge device is down due to a link outage or is administratively disabled. • **Wait for LSP tunnel to Peer** – Cannot find a working tunnel LSP. • **Wait for LDP session to Peer** – The LDP session is not yet ready. • **Wait for PW Up (Wait for remote VC label)**– The device has advertised its VC label binding to the VPLS peer, but has not yet received the peer's VC label binding. • **Wait for PW Up (VC type mismatched)** – A session is not formed because the VC type does not match with its peer's VC type. • **Wait for PW Up (MTU mismatched)** – A session will not be formed and this message will be displayed. The MTU sent to a peer is derived from the router's global setting by the following formula: (system-mtu minus 26 bytes). If a system-mtu value is not configured, a default value of 1500 is sent. • **Wait for PW Up (Wait for LPD session to Peer) -** The LDP session to the peer is down. • **Wait for PW Up (No Label Resource)** - When configuring a new VPLS peer, the maximum amount of VC labels that can be supported may exceed 64K, and cause the configuration to be rejected.The maximum amount of VC labels available for VPLS instances is equal to 64K.
Tnnls in use	The tunnel LSP used to reach the VPLS peer. If VPLS traffic to the peer is load balanced across multiple tunnel LSPs, the tunnel LSPs used to reach the peer are displayed. When load balancing for auto-discovered VPLS peers is enabled for the VPLS instance, "(load balance)" also appears in this line.
LDP session	The state of the LDP session between this device and the VPLS peer.
Local VC lbl	The VC label value locally allocated for this peer for this VPLS instance. Packets forwarded from the VPLS peer to this device are expected to contain this label. This is the label that is advertised to the VPLS peer through LDP.
Remote VC lbl	The VC label allocated by the VPLS peer and advertised to this device through LDP. The device applies this label to outbound MPLS packets sent to the VPLS peer.
Local VC MTU	The MTU value locally configured for this peer.

TABLE 268 Output for the show mpls vpls name command (Continued)

This field...	Displays
Remote VC MTU	The MTU value configured for the remove VPLS peer.
CPU-protection	Indicates whether CPU protection is enabled (ON) or disabled (OFF) for this VPLS instance.
Local Switching	Indicates whether local switching is enabled or disabled for this VPLS instance.

Displaying information about LDP

To display information about LDP, including the router ID and loopback interface in use, enter the **show mpls ldp** command.

```
NetIron(config-lbif-1)#show mpls ldp
Label Distribution Protocol version 1
  LSR ID: 3.3.3.3, using Loopback 2 (deleting it will stop LDP)
  Hello interval: Link 5 sec, Targeted 15 sec
  Hold time value sent in Hellos: Link 15 sec, Targeted 45 sec
  Keepalive interval: 6 sec, Hold time multiple: 6 intervals
  Num VC FEC currently allocated: 0
```

Field definitions for the **show mpls ldp** command are in the section "Displaying the LDP version" on page 1428.

Syntax: **show mpls ldp**

IP over MPLS 35

Overview

One of the benefits that MPLS offers service providers is the ability to take advantage of MPLS traffic engineering capabilities to efficiently utilize the service provider network bandwidth, to control traffic placement, as well as to achieve fast network resiliency. This is accomplished through IP-over-MPLS features.

PowerConnect B-MLXe supports the following IP over MPLS features:

- IP over MPLS
- BGP Shortcut with optional LSP metrics
- IS-IS Shortcuts
- ECMP forwarding for IP over MPLS
- LDP Route Injection
- QoS Mapping Between IP Packets and MPLS
- Using Traffic Engineered LSPs Within an AS
- OSPF Shortcuts
- BGP Shortcut Enhancement

The following sections describe some of the procedures and considerations required when configuring a device to carry IP traffic over an MPLS network:

- "BGP shortcuts" – This feature directs BGP to resolve a route nexthop to a MPLS LSP when one is available.
- "LDP route injection" – This feature allows you to make selected customer routes available though LDP created LSP tunnels.
- "Using traffic-engineered LSPs within an AS" – This section describes how CoS values are determined for packets through an LSP.

BGP shortcuts

In a typical configuration, BGP considers only IP routes when building a routing table. If an MPLS network uses BGP to propagate routes, BGP must consider whether the MPLS tunnels are viable routes. The BGP shortcut feature forces BGP to use an MPLS tunnel as the preferred route to a destination network if one is available. You can also force BGP to include LSP metrics for best-route computations.

When configured on an MPLS edge router, BGP computes routes to destinations available through other edge routers. When BGP determines that a route is available through an edge router that is reachable through an MPLS tunnel, a BGP shortcut directs BGP to place the MPLS tunnel in the routing table as the preferred BGP route.

You can globally enable the BGP shortcut feature and optional inclusion of LSP metrics on a router. With the BGP shortcut feature enabled, the router first attempts to resolve BGP routes with an MPLS tunnel, and can optionally consider LSP metrics. If the BGP attempt at route resolution is unsuccessful, the router defaults to the IPv4 routing table.

Key algorithms

This section describes the behavior of the system in three contexts:

- **Next-hop MPLS disabled:** Only IP routing tables are used to resolve routes for the routing table.
- **Next-hop MPLS enabled:** LSP with a fixed metric of 1 is used to resolve the routes. For routes that cannot be resolved, the system uses the routing table.
- **Next-hop MPLS with LSP metric consideration:** When BGP resolves the next hop with LSP, it uses the LSP metric as the IGP cost for that next hop. If any of the possible paths are through an LSP, then only LSPs are chosen. The IGP cost of each next hop is then compared, and only IGP cost paths with the lowest values are considered for ECMP.

Native IP forwarding

If next-hop MPLS is disabled, BGP uses the default BGP decision process and native IP forwarding to build BGP EMCP routes.

Next-hop MPLS

For each unique BGP next hop, if next-hop MPLS is enabled, BGP first determines if an LSP can be used to resolve the route. If BGP can resolve the route, it does not check the native IP routing table.

For each BGP next hop, if the route is resolved by LSP, then all possible LSPs with the same lowest-metric value are selected. After this selection, BGP internally sets this next hop IGP cost to 1 (rather than the true LSP metric) to force it to be the preferred hop over a hop resolved by native IP.

For each BGP next hop, the IGP cost is compared, and the least-value IGP cost for the next hop or hops are used to install them in the routing table.

When the router installs a BGP route in the RTM, it uses a BGP metric, not the IGP metric (IGP cost.)

Next-hop MPLS comparing LSP metrics

With the option enabled to compare LSP metrics, after BGP resolves a next hop with LSP, it uses the LSP metric as the IGP cost for that next hop. Thereafter, all of the next hops IGP costs are compared, and only the IGP cost paths with the lowest values are considered for ECMP. If any of these paths is an LSP, then only LSP paths are taken.

You have the flexibility to choose a native IP path over an LSP path if they have different BGP next-hops, and the native IP path has a lower IGP cost.

NOTE
Enabling or disabling the LSP metric option takes effect immediately: BGP automatically recalculates the existing BGP routes.

To configure BGP shortcuts and optionally compare LSP metrics, use the **next-hop-mpls** command in BGP configuration mode, as in the following example.

```
NetIron(config)# router bgp
NetIron(config-bgp)# next-hop-mpls compare-lsp-metric
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

For the next-hop-mpls command, when you use the no form with the optional compare-lsp-metric parameter, only this optional parameter is deleted, so the global next-hop MPLS enable remains the same. To disable both the optional LSP-metric compare and the global next-hop MPLS, use the no form of the command but without the optional compare-lsp-metric parameter.

Examples of next-hop MPLS

This section illustrates how to configure a BGP shortcut by enabling next-hop MPLS. It also illustrates the optional parameter—the consideration of LSP metrics:

- Enabling next-hop MPLS (LSP metric becomes fixed at 1)
- Enabling compare-LSP-metric (so IGP metric is compared with user-configurable LSP metric)
- Disabling next-hop MPLS

Enable next-hop MPLS using the next-hop-mpls command, as the following example illustrates. The follow-up show command of the running configuration indicates the global enabling of this feature.

```
NetIron(config-bgp)#next-hop-mpls
NetIron(config-bgp)#show ip bgp config
Current BGP configuration:
router bgp
 local-as 10
 neighbor 10.1.1.2 remote-as 20
 neighbor 20.1.1.2 remote-as 20
 address-family ipv4 unicast
 next-hop-mpls
 exit-address-family
 address-family ipv4 multicast
 exit-address-family
 address-family ipv6 unicast
 exit-address-family
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

Enable the router to use the compare LSP metric. The running configuration reflects the global configuration on one line.

```
NetIron(config-bgp)#next-hop-mpls compare-lsp-metric
NetIron(config-bgp)#show ip bgp config
Current BGP configuration:
router bgp
 local-as 10
 neighbor 10.1.1.2 remote-as 20
 neighbor 20.1.1.2 remote-as 20
 address-family ipv4 unicast
 next-hop-mpls compare-lsp-metric
 exit-address-family
 address-family ipv4 multicast
 exit-address-family
 address-family ipv6 unicast
 exit-address-family
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

This series of examples shows how an IP-only routing table resolution for BGP is affected first by the enabling of next-hop MPLS and then by the enabling of LSP-metric comparison. The tasks for these examples are:

- Specify metrics for three LSPs. The existing LSPs in this example are to2, to22, and to2_sec. As a precondition for this example, their metrics are changed to 10, 20, and 10.
- Enable BGP ECMP, then check the routing table. The destination IP address for this example is 8.8.8.1/32. The routing table shows that native IP-forwarding is used.
- Enable next-hop MPLS and observe the effect on the route to 8.8.8.1/32.
- Enable LSP-metric comparison and note that, because of the metric for LSP to22, it has no effect on the routing table.
- Change the metric for an LSP (to2 in this example).
- Disable LSP-metric compare and check the consequences.
- Disable global next-hop MPLS

Specifying metrics

This step specifies metrics for three LSPs.

```
NetIron(config-bgp)#router mpls
NetIron(config-mpls)#lsp to2
NetIron(config-mpls-lsp-to2)#disable
NetIron(config-mpls-lsp-to2)#to 10.1.1.2
NetIron(config-mpls-lsp-to2)#from 10.1.1.1
NetIron(config-mpls-lsp-to2)#metric 10
NetIron(config-mpls-lsp-to2)#enable
Connecting signaled LSP to2
exit
....
NetIron(config-mpls)#lsp to22
NetIron(config-mpls-lsp-to22)#disable
NetIron(config-mpls-lsp-to22)#to 10.1.1.2
NetIron(config-mpls-lsp-to22)#from 10.1.1.1
NetIron(config-mpls-lsp-to22)#metric 20
NetIron(config-mpls-lsp-to22)#enable
Connecting signaled LSP to22
exit
....
NetIron(config-mpls)#lsp to2_sec
NetIron(config-mpls-lsp-to2_sec)#diable
NetIron(config-mpls-lsp-to2_sec)#to 20.1.1.2
NetIron(config-mpls-lsp-to2_sec)#from 20.1.1.1
NetIron(config-mpls-lsp-to2_sec)#metric 10
NetIron(config-mpls-lsp-to2_sec)#enable
Connecting signaled LSP to2_sec
exit
NetIron(config-mpls)#show mpls lsp
                              Admin Oper  Tunnel   Up/Dn Retry Active
Name           To             State State Intf     Times No.   Path
to2            10.1.1.2       UP    UP    tnl0     1     0     --
to2_sec        20.1.1.2       UP    UP    tnl2     1     0     --
to22           10.1.1.2       UP    UP    tnl1     1     0     --
```

Syntax: [**no**]**metric** <*num*>

Enable BGP ECMP

This example shows BGP ECMP being enabled and the check of the routing table manager (RTM) by the **show ip route** command. The destination for this example is 8.8.8.8/32, and native IP forwarding is in effect.

```
NetIron(config-mpls)#router bgp
NetIron(config-bgp)#maximum 5
NetIron(config-bgp)#show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       1.1.1.1/32         DIRECT          loopback 1  0/0           D    9m46s
2       2.2.3.3/32         DIRECT          loopback 2  0/0           D    9m46s
3       5.5.5.5/32         10.1.1.10       eth 1/1     1/1           S    9m35s
4       8.8.8.1/32         10.1.1.2        eth 1/1     20/0          B    0m1s
        8.8.8.1/32         20.1.1.2        eth 1/2     20/0          B    0m1s
5       8.8.8.2/32         10.1.1.2        eth 1/1     20/0          B    0m1s
        8.8.8.2/32         20.1.1.2        eth 1/2     20/0          B    0m1s
```

Enable next-hop MPLS

In this example, the next-hop MPLS is enabled, and the **show ip route** command is used to check the RTM.

```
NetIron(config-bgp)#next-hop-mpls
NetIron(config-bgp)#show ip route
Total number of IP routes: 4
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       1.1.1.1/32         DIRECT          loopback 1  0/0           D    10m4s
2       2.2.3.3/32         DIRECT          loopback 2  0/0           D    10m4s
3       5.5.5.5/32         10.1.1.10       eth 1/1     1/1           S    9m53s
4       8.8.8.1/32         10.1.1.2        lsp to2     20/0          B    0m1s
        8.8.8.1/32         20.1.1.2        lsp to2_sec 20/0          B    0m1
5       8.8.8.2/32         10.1.1.2        lsp to2     20/0          B    0m1s
        8.8.8.2/32         20.1.1.2        lsp to2_sec 20/0          B    0m1s
```

Syntax: [no] **next-hop-mpls** [**compare-lsp-metric**]

Enable LSP-metric comparison

For this example, LSP-metric comparison is enabled and the consequences are checked in the RTM. In this case, LSPs to2 and to2_sec already provide the best route, so this display does not differ from the example in which next-hop MPLS is enabled. Note that to22 is not displayed because its metric is 20, but the metric of to2 (to the same destination) is only 10 and so represents the chosen LSP.

```
NetIron(config-bgp)# next-hop-mpls compare-lsp-metric
NetIron(config-bgp)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       1.1.1.1/32         DIRECT          loopback 1  0/0           D    11m30s
2       2.2.3.3/32         DIRECT          loopback 2  0/0           D    11m30s
3       5.5.5.5/32         10.1.1.10       eth 1/1     1/1           S    11m19s
4       8.8.8.1/32         10.1.1.2        lsp to2     20/0          B    0m1s
        8.8.8.1/32         20.1.1.2        lsp to2_sec 20/0          B    0m1s
5       8.8.8.2/32         10.1.1.2        lsp to2     20/0          B    0m1s
        8.8.8.2/32         20.1.1.2        lsp to2_sec 20/0          B    0m1s
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

Changing the metric for an LSP

In the next example, the metric for LSP to2 is changed to a value (20) that causes the system to remove it from the routing table, so only LSP to2_sec to 8.8.8.1/32 remains. This output illustrates this result.

```
NetIron(config-mpls)# lsp to2
NetIron(config-mpls-lsp-to2)# disconnect
Disconnecting signaled LSP
NetIron(config-mpls-lsp-to2)# metric 20
NetIron(config-mpls-lsp-to2)# enable
Connecting signaled LSP to2
NetIron(config-mpls)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       1.1.1.1/32         DIRECT          loopback 1  0/0           D    12m23s
2       2.2.3.3/32         DIRECT          loopback 2  0/0           D    12m23s
3       5.5.5.5/32         10.1.1.10       eth 1/1     1/1           S    12m12s
4       8.8.8.1/32         20.1.1.2        lsp to2_sec 20/0          B    0m6s
5       8.8.8.2/32         20.1.1.2        lsp to2_sec 20/0          B    0m6s
```

Disabling LSP-metric compare and checking the consequences

For the last example related to next-hop MPLS, disable LSP-metric compare using the **no** form of the **next-hop-mpls** command and include the **compare-lsp-metric** option.

NOTE

When you use the **no** form with the optional **compare-lsp-metric** parameter for the **next-hop-mpls** command, only this optional parameter is deleted, so global next-hop-mpls enable remains the same. To disable both the optional LSP-metric compare and the global next-hop-mpls, use the **no** form of the **next-hop-mpls** command without the optional parameter.

Because global next-hop MPLS remains enabled and the LSP metrics are no longer a factor, all the LSPs are displayed in the routing table since BGP considers them to have equal cost.

```
NetIron(config-bgp)# no next-hop-mpls compare-lsp-metric
NetIron(config-bgp)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       1.1.1.1/32         DIRECT          loopback 1  0/0           D    12m58s
2       2.2.3.3/32         DIRECT          loopback 2  0/0           D    12m58s
3       5.5.5.5/32         10.1.1.10       eth 1/1     1/1           S    12m47s
4       8.8.8.1/32         10.1.1.2        lsp to2     20/0          B    0m1s
        8.8.8.1/32         10.1.1.2        lsp to22    20/0          B    0m1s
        8.8.8.1/32         20.1.1.2        lsp to2_sec 20/0          B    0m1s
5       8.8.8.2/32         10.1.1.2        lsp to2     20/0          B    0m1s
        8.8.8.2/32         10.1.1.2        lsp to22    20/0          B    0m1s
        8.8.8.2/32         20.1.1.2        lsp to2_sec 20/0          B    0m1s
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

Disabling global next-hop MPLS

Disable global next-hop MPLS and check the RTM to see that native IP-forwarding has been restored.

```
NetIron(config-bgp)# no next-hop-mpls
NetIron(config-bgp)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       1.1.1.1/32         DIRECT          loopback 1  0/0           D    9m46s
2       2.2.3.3/32         DIRECT          loopback 2  0/0           D    9m46s
3       5.5.5.5/32         10.1.1.10       eth 1/1     1/1           S    9m35s
4       8.8.8.1/32         10.1.1.2        eth 1/1     20/0          B    0m1s
        8.8.8.1/32         20.1.1.2        eth 1/2     20/0          B    0m1s
5       8.8.8.2/32         10.1.1.2        eth 1/1     20/0          B    0m1s
        8.8.8.2/32         20.1.1.2        eth 1/2     20/0          B    0m1s
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

LDP route injection

An MPLS edge router is typically connected to a customer network that is not configured for MPLS. If the edge router is then connected to the MPLS core through LSP tunnels that have been created by LDP, only routes to the loopback address of the edge router are available for routing through the LSP tunnels. In practice, this means that routes to and from the customer network are unavailable to the MPLS network.

The LDP route injection feature allows you to make routes available from the customer network through LSPs that have been created by LDP. You can filter routes that you want to allow through the MPLS network using an ACL, and then apply that ACL to the **advertise-labels for** command. The routes injected will then be accessible over the MPLS network.

To direct the device to inject non-loopback routes into LDP while restricting the routes injected through reference to an ACL, enter the following command.

```
NetIron(config)# router mpls
NetIron(config-mpls)# ldp
NetIron(config-mpls-ldp)# advertise-labels for 30
```

Syntax: advertise-labels for <*access-list*>

The <*access-list*> variable refers to the number of the access list that filters for the routes you want to use for label binding.

Considerations when using LDP route injection

1. You can directly change the LDP route injection filter without deleting a previously configured one. The change automatically applies and triggers LDP route re-injections.
2. Any change to a referenced ACL automatically applies to LDP route injection filtering and triggers LDP route re-injection.
3. If no LDP route injection filter is configured, by default LDP acquires all local loopback addresses.
4. If the ACL referenced by the LDP route injection filter is not configured, it is an implicit deny. All local routes are denied.
5. Both number-based and name-based ACLs can be used. Because only prefix-based filtering is applied, use of a standard ACL is preferred.
6. The LDP route injection filter is only applied on local route injection. Learned remote binding is not filtered.

LDP route injection example

This example describes how to use LDP route injection to inject routes 1.2.2.2/32 and 5.5.5.2/32 into the LDP label information database.

1. the **show ip interface** command displays IP addresses of loopback interfaces in router MLXe-1.

```
MLXe-1# show ip interface
Interface       IP-Address      OK?  Method    Status    Protocol  VRF
eth 1/1         20.0.0.1        YES  NVRAM     up        up        default
eth 1/2         120.0.0.1       YES  NVRAM     up        up        default
loopback 3      3.3.3.3         YES  NVRAM     up        up        default
loopback 5      5.5.5.5         YES  manual    up        up        default
```

2. By default, the LDP label information database only contains labels learned for IP addresses of loopback interfaces, as demonstrated in this example, where only prefixes 3.3.3.3/32 and 5.5.5.5/32 are displayed by the **show mpls ldp database** command.

```
MLXe-1# show mpls ldp database
Session 3.3.3.3:0 - 5.5.5.2:0
 Downstream label database:
   Label     Prefix              State
   1024      3.3.3.3/32          Retained
 Upstream label database:
   Label     Prefix
   3         3.3.3.3/32
   3         5.5.5.5/32
```

3. The **show ip route** command displays routes available to ports on router MLXe-1.

```
MLXe-1# show ip route
Total number of IP routes: 9
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metr
       Destination        Gateway         Port        Cost       Type
1      1.2.2.2/32         20.0.0.2        eth 1/1     1/1        S
2      3.3.3.3/32         DIRECT          loopback 3  0/0        D
3      5.5.5.0/24         20.0.0.2        eth 1/1     1/1        S
4      5.5.5.1/32         20.0.0.2        eth 1/1     1/1        S
5      5.5.5.2/32         20.0.0.2        eth 1/1     110/2      O
6      5.5.5.5/32         DIRECT          loopback 5  0/0        D
7      5.5.6.2/32         20.0.0.2        eth 1/1     1/1        S
8      20.0.0.0/24        DIRECT          eth 1/1     0/0        D
9      120.0.0.0/24       DIRECT          eth 1/2     0/0        D
```

4. In this example, a filter is configured to inject route 1.2.2.2/32.

```
MLXe-1(config)# access-list 30 permit 1.2.2.2/32
MLXe-1(config)# router mpls
MLXe-1(config-mpls)# ldp
MLXe-1(config-mpls-ldp)# advertise for 30
```

5. As shown, the 1.2.2.2/32 has been injected into the LDP Label information database.

```
MLXe-1# show mpls ldp database
Session 3.3.3.3:0 - 5.5.5.2:0
 Downstream label database:
   Label     Prefix              State
 Upstream label database:
   Label     Prefix
   3         1.2.2.2/32
```

6. In this example a second filter is configured to inject route 5.5.5.2/32.

```
MLXe-1(config)#access-list 30 permit 5.5.5.2/32
```

7. As shown, route 5.5.5.2/32 has been injected into the LDP label information database.

```
MLXe-1(config)# show mpls ldp database
Session 3.3.3.3:0 - 5.5.5.2:0
 Downstream label database:
   Label     Prefix              State
 Upstream label database:
   Label     Prefix
   3         1.2.2.2/32
   3         5.5.5.2/32
```

Displaying routes through LSP tunnels

Once a network has been enabled to allow routes through LSP tunnels, the routes will appear in the IP routing table. In the following example, the **show ip route** command displays a table that contains routes through LSP tunnels. In this example, routes 7 - 8 and 10 - 14 are LDP tunnels.

```
NetIron# show ip route
Total number of IP routes: 1027
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost -
Dist/Metric
        Destination        Gateway         Port        Cost       Type
1       1.1.1.1/32         DIRECT          loopback 1  0/0        D
2       1.1.2.1/32         DIRECT          loopback 2  0/0        D
3       1.1.3.1/32         DIRECT          loopback 3  0/0        D
4       2.2.2.2/32         11.0.0.2        eth 1/1     110/10     O
5       3.3.3.3/32         11.0.0.2        eth 1/1     110/12     O
        3.3.3.3/32         11.8.0.2        eth 1/4     110/12     O
6       4.4.4.4/32         11.8.0.2        eth 1/4     110/10     O
7       5.5.1.5/32         5.5.5.5         lsp(LDP)    200/0      B
8       5.5.3.5/32         5.5.5.5         lsp(LDP)    200/0      B
9       5.5.5.5/32         11.0.0.2        eth 1/1     110/13     O
        5.5.5.5/32         11.8.0.2        eth 1/4     110/13     O
10      6.6.1.6/32         6.6.6.6         lsp(LDP)    200/0      B
11      6.6.2.6/32         6.6.6.6         lsp(LDP)    200/0      B
12      6.6.3.6/32         6.6.6.6         lsp(LDP)    200/0      B
13      6.6.4.6/32         6.6.6.6         lsp(LDP)    200/0      B
14      6.6.5.6/32         6.6.6.6         lsp(LDP)    200/0      B
15      6.6.6.6/32         11.0.0.2        eth 1/1     110/14     O
        6.6.6.6/32         11.8.0.2        eth 1/4     110/14     O
```

Using traffic-engineered LSPs within an AS

In addition to traffic destined to travel outside an AS, Dell routers can forward internal AS traffic into LSP tunnels. This feature allows you to configure a signalled LSP to serve as a shortcut between nodes in an AS. In a shortcut LSP, OSPF includes the LSP in the SPF calculation. If OSPF determines that the LSP shortcut is the best path to a destination, it installs a route into the IP routing table, specifying the LSP tunnel interface as the outbound interface, as well as the coxt of the LSP. Only LSPs configured to router IDs can be considered as shortcuts. If the LSP goes down or is administratively disabled, the LSP tunnel route is removed from the main routing table.

The cost of the LSP is the user-configured metric for the LSP. If there is no user-configured metric, the underlying IP cost of the LSP is used. For example, if the IP cost of the best underlying path between two routers is 2, and there is an LSP configured between these two routers, the cost of the LSP is 2. Once an LSP is used as a next hop for a destination, the cost of the LSP can be used to calculate other destinations that can use the LSP egress node as next hop. This allows traffic for addresses downstream from the LSP egress node (including prefixes of the egress node) to use the LSP shortcut.

If OSPF is already using an LSP tunnel route to an Area Border Router (ABR), all inter-area routes through that ABR use the LSP as the next hop, provided there are no other better paths to the destination (paths through other ABRs). An LSP to a destination outside an area is not used by OSPF in the calculation of inter-area routes.

Only signalled LSPs can be used as OSPF shortcuts. RSVP packets, used to establish and maintain signalled LSPs, are never forwarded into LSP tunnels.

Refer to “Creating OSPF shortcuts over an LSP tunnel” on page 1585 for more information.

IS-IS shortcuts over an LSP tunnel

Refer to "IS-IS shortcuts" on page 1586 for details about creating IS-IS shortcuts over an LSP tunnel.

Creating OSPF shortcuts over an LSP tunnel

This feature allows you to forward traffic to destinations within an OSPF routing domain through an LSP tunnel, which optimizes available bandwidth by choosing LSPs where multiple paths exist to the same OSPF destination. When an LSP is configured as an OSPF shortcut, OSPF includes the LSP in the SPF calculation. If OSPF determines that the LSP shortcut is the best path to a destination, it adds a route to the IP routing table, specifying the LSP tunnel interface as the outbound interface, along with the cost of the LSP. Only LSPs configured to router ID are considered as shortcuts. If the LSP goes down or is administratively disabled, or the **shortcuts ospf** command is removed from the configuration, the LSP tunnel route is removed from the main routing table.

LSPs used for this feature must originate and terminate within the same OSPF area. When configured, OSPF directs routes that are reachable from the egress router of a shortcut-enabled LSP to an LSP tunnel as the outgoing interface.

To configure this feature, point the LSP to the router ID of the egress router where traffic will be forwarded. You must also configure the LSP with the **shortcuts ospf** command.

The following configuration of LSP "tunnel1" specifies the egress router with a router ID of 2.2.2.2 and enables it for OSPF shortcuts.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp tunnel1
NetIron(config-mpls-lsp)# to 2.2.2.2
NetIron(config-mpls-lsp)# shortcuts ospf
NetIron(config-mpls-lsp)# enable
```

Syntax: [no] **shortcuts ospf**

This feature points OSPF routes to routes from the configured egress router of the LSP tunnel. By way of the LSP interface, the ingress router points to routes on the egress router (including downstream external or summary routes). To view these routes, enter the **show ip route** command as shown in the following example.

```
NetIron# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D: Connected I:ISIS S: Static R:RIP O: OSPF; Cost - Dist/Metri
       Destination        Gateway        Port          Cost       Type

1      2.2.2.0/24         2.2.2.2        lsp tunnel1   110/10     O2
2      5.5.5.0/24         11.1.1.2       eth 1/1       110/2      O
3      15.15.15.15/32     3.3.3.3        lsp l1        110/10     O2
4      78.0.0.0/8         11.1.1.2       eth 1/1       110/10     O2
5      192.85.1.0/24      11.1.1.2       eth 1/1       110/2      O
```

In this example, Type "O2" routes are OSPF routes from outside the OSPF area.

You can set the next hop for a static route to the egress router of an LSP tunnel if the destination route is contained in the MPLS routing table, as described in "Static route to an LSP tunnel interface" on page 727.

IS-IS shortcuts

This section describes IS-IS shortcuts and how to configure them on an MPLS router with traffic engineering (TE) capabilities.

Overview

The IS-IS Shortcuts feature enables an MPLS TE path (LSP tunnel) to serve as a shortcut through the network to a destination based on the cost of the path (metric). Traffic is forwarded through the LSP tunnel to destinations within the IS-IS routing domain. This feature helps optimize available bandwidth by choosing paths using LSPs where multiple paths exist to the same destination.

When IS-IS shortcuts are enabled on an LSP tunnel, IS-IS includes the LSP in the SPF calculation. If IS-IS determines that the LSP shortcut is the best path to a destination, it adds the route to the IP routing table, specifying the LSP tunnel interface as the outbound interface, including the cost of the LSP. Only LSPs configured to a router ID are considered as shortcuts. If the LSP goes down or is administratively disabled, or if the **shortcuts isis** command is removed from the configuration, the IS-IS LSP tunnel routes are removed from the main routing table.

Determining the cost of an IS-IS shortcut

IS-IS uses the following information to determine the cost of an IS-IS shortcut:

- The ***announce metric***, if announce is enabled
- If announce is not enabled, IS-IS uses the LSP metric configured under the LSP configuration mode, for example:

  ```
  NetIron(config-mpls-lsp)#metric <value>
  ```

- If no LSP metric is configured, IS-IS uses the native IGP cost, plus or minus the ***relative metric***
- If there is no relative metric, IS-IS uses the native IGP cost

The announce metric and relative metric are described in detail in the following sections.

The announce metric

When IS-IS shortcuts are enabled on an LSP tunnel, the MPLS router does not announce (advertise) the IS-IS shortcuts unless specifically configured to do so. When announce is enabled, you can optionally specify an announce metric, which is used to compute the LSP cost of the IS-IS shortcut. If an announce metric is not explicitly configured, IS-IS uses a default metric value of 10.

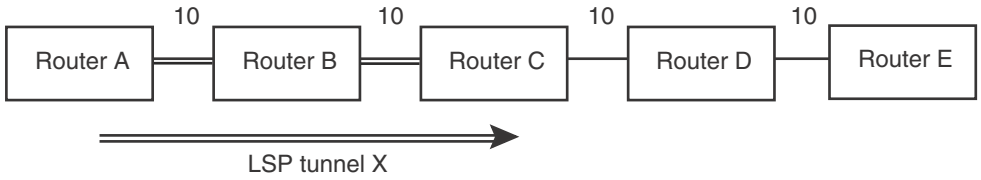
To configure an announce metric for IS-IS shortcuts, refer to "“Configuring the announce metric” on page 1589".

The relative metric

When announce is not enabled and an LSP metric is not explicitly configured under the LSP configuration mode of the CLI, the ***relative metric*** is used to compute the LSP cost, which is the native IGP cost, plus or minus the relative metric.

The relative metric is optionally specified when IS-IS shortcuts are enabled, and is used to make an LSP tunnel less or more preferred over other paths. The default relative metric value is zero (0), but can be configured to be a positive or negative number. A positive number disables an LSP tunnel from participating in the SPF calculation. A negative number ensures that the LSP tunnel is preferred over native IGP paths in the SPF calculation. Figure 201 shows an example of this configuration.

FIGURE 201 SPF calculation adjustment using the relative metric



In this example, if there are no IS-IS shortcuts, Router A adds routes in the routing table for routers C, D, and E with the metrics 20, 30, and 40, respectively. If an IS-IS shortcut is configured on LSP tunnel X and the relative metric is –5 (minus 5), router A installs the same routes in the routing table with the metrics 15, 25, and 35, respectively, over the LSP tunnel X.

To configure the device to use a relative metric value other than 0, refer to "Configuring the relative metric" on page 1589.

Why LSP tunnels may be excluded from the SPF calculation

LSP tunnels may be excluded in SPF calculations in the following cases:

- The system did not find mapping between the LSP tunnel destination (the To address) and the IS-IS system ID.
- There is no IS-IS native route to the LSP tunnel destination.
- The IS-IS native route has a better metric than the LSP tunnel.
- Another shortcut has a better metric than the LSP tunnel.

Configuration notes

Consider the following configuration notes:

- IS-IS shortcuts require MPLS and IS-IS Traffic Engineering (TE) to be enabled. For details about IS-IS TE, refer to "IS-IS Link State Protocol data units with TE extensions for MPLS interfaces" on page 1299. To enable IS-IS TE, refer to "Enabling IS-IS LSPs with TE extensions for MPLS interfaces" on page 1322.
- IS-IS does not use an LSP tunnel as a shortcut if the To address of the tunnel is not the router ID of the destination router.
- Where multiple IS-IS shortcuts have the same cost, IS-IS installs LSP tunnel-based ECMP routes.

Configuration tasks

It is recommended that you perform the configuration tasks in the order listed in Table 269.

TABLE 269 Configuration tasks for IS-IS shortcuts

Configuration task		Default behavior	See...
1	Enable IS-IS shortcuts	Disabled	"Enabling and disabling IS-IS shortcuts"
2	Optionally enable announce on the LSP	Disabled	"Enabling IS-IS shortcut advertisements"
3	If announce is enabled, optionally configure the announce metric	When announce is enabled, the system uses either the default metric value of 10, or the explicitly-configured announce metric value.	"Configuring the announce metric"
4	Optionally configure the relative metric	The default value is 0 (zero).	"Configuring the relative metric"

After performing the configuration steps listed in Table 269, you can observe the IS-IS routes that use IGP shortcuts. For more information, refer to "Show command support" on page 1593.

Enabling and disabling IS-IS shortcuts

To enable IS-IS shortcuts on an LSP tunnel, enter commands such as the following, starting at the MPLS level of the CLI.

```
NetIron MLX(config-mpls)# lsp tomu3
NetIron MLX(config-mpls-lsp-tomu3)# shortcuts isis level2
NetIron MLX(config-mpls-lsp-tomu3)# enable
Connecting signaled LSP tomu3
```

These commands enable IS-IS shortcuts on the **tomu3** LSP tunnel.

Syntax: [**no**] **shortcuts isis level1** | **level 2**

Enter the **no** form of the command to disable IS-IS shortcuts.

The **level1** or **level2** keyword is required and indicates the level of IS-IS routing enabled on the device. The levels are:

- level1 – A level1 router routes traffic only within the area that includes the router. To forward traffic to another area, a level1 router sends the traffic to the nearest level2 router.
- level2 – A level2 router routes traffic between areas within a domain.

NOTE
The IS-IS traffic engineering level should match the shortcut level configuration.

Enabling IS-IS shortcut advertisements

When announce is enabled, the tunnel information is advertised in an IS neighbor TLV, which is stored in the IS-IS database.

To enable announce, enter the following command on an LSP that is not yet enabled.

```
NetIron MLX(config-mpls-lsp-tomu3)# shortcuts isis level2 announce
```

If the tunnel is enabled, disable it before enabling announce, then re-enable the tunnel. For example.

```
NetIron MLX(config-mpls-lsp-tomu3)# disable
Disconnecting signaled LSP tomu3
NetIron MLX(config-mpls-lsp-tomu3)# shortcuts isis level2 announce
NetIron MLX(config-mpls-lsp-tomu3)# enable
Connecting signaled LSP tomu3
```

These commands enable the system to advertise IS-IS shortcuts. Since an announce metric is not explicitly specified in this example, IS-IS uses the default announce metric of 10. To configure an announce metric other than 10, refer to "Configuring the announce metric".

Syntax: [no] **shortcuts isis level1** | **level2 announce**

Enter the **no** form of the command to disable advertisement of IS-IS shortcuts. IS-IS shortcuts are still enabled, but will no longer be advertised in the IS-IS database.

Configuring the announce metric

The announce metric is described in "The announce metric" on page 1586.

To configure an announce metric, enter a command such as the following at the MPLS LSP level of the CLI.

```
NetIron MLX(config-mpls-lsp-tomu3)# shortcuts isis level2 announce
announce-metric 20
```

Syntax: [no] **shortcuts isis level1** | **level2 announce announce-metric** <*num*>

Enter the **no** form of the command to return to the default announce metric value of 10. IS-IS shortcuts will still be enabled, however the **no** form of the command simply reverts to the default announce metric.

For <*num*>, enter a value from 1 – 16777215. The default is 10.

The announce metric is displayed in the output of the **show isis shortcuts** command. If the LSP tunnel is not announced, a – (dash) is displayed in the announce metric field.

Configuring the relative metric

The relative metric is described in "The relative metric" on page 1586.

If announce is not enabled and a metric is not explicily configured under the LSP configuration mode of the CLI, the ***relative metric*** is used to compute the shortcut cost.

To configure the relative metric, enter a command such as the following at the MPLS LSP level of the CLI.

```
NetIron MLX(config-mpls-lsp-tomu3)# shortcuts isis level2 relative-metric -5
```

This command sets the relative metric value to -5. The LSP cost is determined by subtracting 5 from the native IGP cost to reach the tunnel destination. Using this example, if the native IGP cost is 10, the relative metric value -5 sets the LSP cost to 5.

NOTE
The shortcut cost will never be a value less than 1. For example, if the native IGP cost is 10 and the relative metric is -15, the shortcut cost will be 1, not -5.

Syntax: [no] **shortcuts isis level1** | **level2 relative-metric +** | **–** <*num*>

Enter the **no** form of the command to return to the default native IGP path metric. IS-IS shortcuts will still be enabled. The **no** form of the command simply removes the relative-metric value from the configuration.

The **+** or **–** sign is required. **+** denotes a positive number. – denotes a negative number.

For <*num*>, enter a value from 1 – 16777215. The default is 0 (zero).

The metric used in the SPF calculation is displayed in the output of the **show isis shortcuts** command. If the LSP tunnel is not used in the SPF calculation, a – (dash) is displayed in the SPF metric field.

Example configurations

This section includes example configurations and relevant **show** command outputs both before and after IS-IS shortcuts are enabled.

The following display shows an IS-IS route configuration *before* IS-IS shortcuts are installed.

NOTE
For a description of the output fields, refer to "Displaying the IP route table" on page 763 in 19, "Configuring IP".

```
PowerConnect(config-mpls-lsp-tomu3)# show ip route isis
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime

1       0.0.0.0/0          10.1.1.2        eth 1/1     115/21        IL2  0m23s
2       20.1.1.0/24        10.1.1.2        eth 1/1     115/20        IL2  0m23s
3       30.1.1.0/24        10.1.1.2        eth 1/1     115/10        IL2  0m23s
4       40.1.1.1/32        10.1.1.2        eth 1/1     115/10        IL2  0m23s
5       40.2.2.2/32        10.1.1.2        eth 1/1     115/10        IL2  0m23s
6       100.1.0.0/16       DIRECT          drop        115/10        IL1  3m37s
7       200.1.1.1/32       10.1.1.2        eth 1/1     115/20        IL2  0m23s
```

The following example shows IS-IS shortcut configuration.

```
NetIron MLX(config-mpls)# lsp tomu3
NetIron MLX(config-mpls-lsp-tomu3)# metric 1
NetIron MLX(config-mpls-lsp-tomu3)# shortcuts isis level2
NetIron MLX(config-mpls-lsp-tomu3)# enable
Connecting signaled LSP tomu3
```

The following display shows the IS-IS route configuration *after* the shortcut configuration is applied. The bold text indicates that the routes are now using shortcuts. Compare this output with the output generated before the shortcut configuration was applied.

```
mu1(config-mpls)# show ip route isis
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime

1       0.0.0.0/0          30.1.1.1        lsp tomu3   115/2         IL2  0m2s
2       20.1.1.0/24        30.1.1.1        lsp tomu3   115/11        IL2  0m2s
3       30.1.1.0/24        30.1.1.1        lsp tomu3   115/1         IL2  0m2s
4       40.1.1.1/32        30.1.1.1        lsp tomu3   115/1         IL2  0m2s
5       40.2.2.2/32        30.1.1.1        lsp tomu3   115/1         IL2  0m2s
6       100.1.0.0/16       DIRECT          drop        115/10        IL1  3m54s
7       200.1.1.1/32       30.1.1.1        lsp tomu3   115/1         IL2  0m2s
```

The following example shows an IS-IS shortcut configuration with route advertisements enabled.

```
NetIron MLX(config-mpls)# lsp tomu3
NetIron MLX(config-mpls-lsp-tomu3)# disable
Disconnecting signaled LSP tomu3
NetIron MLX(config-mpls-lsp-tomu3)# shortcuts isis level2 announce
NetIron MLX(config-mpls-lsp-tomu3)# enable
Connecting signaled LSP tomu3
```

In the output for this configuration, the bold text indicates that the device uses the extended TLV fields to advertise the shortcut in an IS adjacency TLV. If route advertisements are not enabled, this text would not appear in the output.

```
PowerConnect(config-mpls)# show isis database mu1.00-00 detail
IS-IS Level-2 Link State Database
LSPID                                 Seq Num     Checksum  Holdtime  ATT/P/OL
mu1.00-00*                            0x00000010  0xd938    35        1/0/0
   Area Address:  47
   NLPID:  IPv6 IP
   Hostname:  mu1
   TE Router ID: 10.1.1.1
   Metric:  10       IP-Extended 10.1.1.0/24         Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 13.1.1.0/24         Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.2.0/24        Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.3.0/24        Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.4.0/24        Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.5.0/24        Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.6.0/24        Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.7.0/24        Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.8.0/24        Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.9.0/24        Up: 0 Subtlv: 0
   Metric:  1        IP-Extended 200.1.10.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 100.1.0.0/16        Up: 0 Subtlv: 0
   Metric:  10       IPv6 Reachability 1000::/32      Up: 0 Subtlv: 0
   Metric:  10       IPv6 Reachability 2000::/32      Up: 0 Subtlv: 0
   Metric:  10       IS-Extended mu1.02
     Admin Group: 0x00000000
     Interface IP Address: 13.1.1.1
     Link BW: 10000000 kbits/sec
     Reservable BW: 10000000 kbits/sec
     Unreserved BW:
       [0]   10000000 kbits/sec  [1]   10000000 kbits/sec
       [2]   10000000 kbits/sec  [3]   10000000 kbits/sec
       [4]   10000000 kbits/sec  [5]   10000000 kbits/sec
       [6]   10000000 kbits/sec  [7]   10000000 kbits/sec
     Admin Group: 0x00000000
     Interface IP Address: 10.1.1.1
     Neighbor IP Address: 10.1.1.2
     Link BW: 10000000 kbits/sec
     Reservable BW: 8000000 kbits/sec
     Unreserved BW:
       [0]    8000000 kbits/sec  [1]    8000000 kbits/sec
       [2]    8000000 kbits/sec  [3]    8000000 kbits/sec
       [4]    8000000 kbits/sec  [5]    8000000 kbits/sec
       [6]    8000000 kbits/sec  [7]    8000000 kbits/sec
    Metric:  10       IS-Extended mu3.00
```

Clearing IS-IS shortcuts

When you clear IS-IS shortcuts, IS-IS attempts to remap the LSP To address to IS-IS system ID. Clearing shortcuts is useful when the mapping between the To address and System ID must be refreshed once the LSP tunnel is being used in the SPF calculation.

NOTE
This is not a common operation.

To clear IS-IS shortcuts from the configuration, use one of the following CLI commands at any level of the CLI:

- **clear isis shortcut** – This command clears all IS-IS shortcuts from the configuration.

- **clear isis shortcut lsp** <*lsp-name*> **–** This command clears IS-IS shortcuts for the specified LSP.

Syntax: **clear isis shortcut** [**lsp** <*lsp-name*>]

Show command support

Use the following **show** commands to display information about IS-IS shortcuts:

- **show isis shortcuts** – Displays information about all IS-IS shortcuts configured on the router.
- **show isis shortcuts lsp** <*lsp-name*> – Displays information about all IS-IS shortcuts configured for a specified LSP.
- **show isis shortcuts detail** – Displays detailed information about all IS-IS shortcuts that are UP, such as the system ID and matching To address of the tunnel, configured metric values, and the time period for which the LSP has been an IS-IS shortcut.
- **show isis shortcuts lsp** <*lsp-name*> **detail** – Displays detailed information about all IS-IS shortcuts for a specified LSP, such as the system ID and matching To address of the tunnel, configured metric values, and the time period for which the LSP has been an IS-IS shortcut.
- **show isis** – Enhanced output indicates whether or not IS-IS shortcuts are configured, the number of shortcuts configured, how many are UP, and how many are advertised.
- **show isis debug** – Shows debugging information for IS-IS shortcuts. For more information, refer to the *PowerConnect B-MLXe Diagnostic Reference*.

NOTE
Only LSPs that are UP (administratively and operationally enabled in the MPLS domain) are kept in the database and displayed in the **show** command outputs. LSPs that are down are not kept in the database and are not displayed in the command outputs.

Displaying general information about IS-IS shortcuts

The **show isis shortcuts** command displays information about all IS-IS shortcuts configured on the device.

```
PowerConnect# show isis shortcuts
Configured: 3, Up: 2, Announced: 1
Name                To                 Metric                Announce    Tunnel
                                   (SPF/Announce)                          Intf
lsp tomu2         40.1.1.1             10/-                  No          tnl1
lsp tomu3         30.1.1.1             -/-                   Yes         tnl2
lsp toolong       200.1.1.1            10/10                 Yes         tnl3
toreachmu3
```

Syntax: **show isis shortcuts** [**lsp** <*lsp-name*>]

The optional **lsp** parameter displays information about IS-IS shortcuts for a specific LSP.

Table 270 describes the fields shown in this output.

TABLE 270 Output for the **show is-is shortcuts** command

This field...	Displays
Configured	The number of IS-IS shortcuts configured.
Up	The number of IS-IS shortcuts that are UP.

TABLE 270 Output for the **show is-is shortcuts** command (Continued)

This field...	Displays
Announced	The number of IS-IS shortcuts that are advertised.
Name	The name of the IS-IS shortcut. If the name is longer than 11 characters, it wraps to the next line.
To	The LSP endpoint address.
Metric (SPF or Announce)	The metric used in the SPF calculation or the metric used in the advertisement of the IS adjacency TLV. The SPF metric can be one of the following: • The metric configured at the MPLS LSP configuration level. • The native IGP metric plus or minus (+ or -) the relative metric configured with the **shortcuts isis** command. • The native IGP metric. • A dash (-) denotes that the tunnel is not used in SPF calculations. The Announce metric can be one of the following: • 10 (the default announce metric) • The metric configured with the **announce-metric** keyword • A dash denotes that the tunnel is not used in the IS adjacency TLV advertisement.
Announce	Indicates whether or not IS-IS shortcuts are advertised: • Yes – IS-IS shortcuts are advertised. • No – IS-IS shortcuts are not advertised.
Tunnel Intf	The tunnel index of the LSP. This is assigned by MPLS whenever an LSP is created.

Displaying detailed information about IS-IS shortcuts

The **show isis shortcuts detail** command displays detailed information about IS-IS shortcuts, including:

- The system ID and matching To address of the tunnel
- Configured metric values
- How long the LSP has been an ISIS shortcut

The following shows output from this command.

```
PowerConnect# show isis shortcuts lsp tomu2 detail
lsp tomu2
 To 40.1.1.1, Used by SPF (10), Not Announced
  LSP metric: 10, Relative Metric: -, Announce Metric: -
  ISIS System Id for 40.1.1.1 is mu2.00-00
  Not announced due to configuration
  Last notification from MPLS received 0h0m35s ago.
```

NOTE
The LSP name in this output is not wrapped.

Syntax: show isis shortcuts detail | **lsp** <*lsp-name*> **detail**

The optional **lsp** parameter displays detailed information about IS-IS shortcuts for a specific LSP.

Table 270 defines the fields shown in this output.

TABLE 271 Output for the **show isis shortcuts detail** command

This field...	Displays
<*name*>	The name of the IS-IS shortcut.
To	This line contains the following information: • The LSP endpoint address • Whether or not this LSP is used in the SPF calculation. This field displays either **Used by SPF** or **Not Used by SPF**. • Whether or not the announce metric is used
LSP metric	This field displays one of the following: • The metric value configured at the MPLS LSP configuration level of the CLI • A dash (-), which denotes that the LSP metric is not configured.
Relative metric	This field displays one of the following: • The relative metric value configured with the **shortcuts isis** command • A dash (-), which denotes that the relative metric is not configured.
Announce metric	This field displays one of the following: • The announce metric value configured with the **shortcuts isis** command • A dash (-) which denotes that the announce metric is not configured.
ISIS System Id	The matching IS-IS system ID for the LSP endpoint.
Not used by SPF due to	If the tunnel is not used by SPF, one of the following reasons is noted: • **Not used by SPF due to no ISIS system-id mapping to** <*router-ID*>. No mapping exists between the LSP tunnel destination and the IS-IS System ID. • **Not used by SPF due to no ISIS native route to LSP** <*tunnel destination*>. There is no IS-IS native route to the LSP tunnel destination. • **Not used by SPF due to ISIS alternate path preferred to this tunnel**. An alternate path has a better metric than the LSP tunnel.
Not announced due to configuration	Indicates that announce is not configured.
Last notification from MPLS received	The last time (in hours, minutes, seconds) a status notification was received from MPLS.

Displaying IS-IS shortcut statistics

The **show isis** command output includes the following information about IS-IS shortcuts:

- Whether or not IS-IS shortcuts are enabled
- The number of IS-IS shortcuts configured
- How many IS-IS shortcuts are UP
- How many IS-IS shortcuts are advertised

The information is displayed at the bottom of the **show isis** display output. For example:

```
NetIron# show isis
(truncated for brevity)...
ISIS Shortcuts: 20 configured, 10 are up, and 10 are announced
```

Or

```
NetIron# show isis
(truncated for brevity)...
No isis shortcuts configured
```

ECMP forwarding for IP over MPLS

ECMP hardware forwarding is supported for IP over MPLS packets when an outgoing interface is configured as a physical port *and* a VE interface, or configured on an MPLS tunnel. When multiple routes use ECMP to reach a destination, hardware ECMP is automatically enabled. ECMP load sharing for IP over MPLS is supported for 2-8 tunnels, with a default of 4 tunnels.

ECMP hardware forwarding is not supported for dynamic mode. Hitless upgrade for ECMP hardware forwarding is not supported.

For ECMP hardware forwarding, all outgoing interface paths must configured in the same VRF, and must belong to an MPLS tunnel. A hash value is computed for a packet when it is received by XPP. XPP uses the hash value to select a PRAM that forwards the packet to the destination. An ECMP PRAM block consists of 8 PRAMs. The hash value for each outgoing packet on a customer edge router interface is calculated based on source MAC address, destination MAC address, VLAN ID, source IP address, destination IP address, IP protocol field, TCP or UDP source port, and TCP or UDP destination port.

The hash value for each incoming packet on the route target is calculated based on the source MAC address, destination MAC address, VLAN ID, source IP address, destination IP address, IP protocol field, TCP or UDP source port, TCP or UDP destination port, and a VC label for an MPLS packet.

QoS mapping between IP packets and MPLS

The 3-bit EXP field in the MPLS header can be used to define a Class of Service (CoS) value for packets the traverse an LSP. The CoS value specifies a priority for MPLS packets.

There are two ways that a CoS value can be applied to packets that traverse an MPLS network through an LSP:

- A CoS value is manually configured for the LSP, as described in "Setting a Class of Service value for the LSP" on page 1341. This is the default operation.
- No CoS value is set for an LSP, and the Type of Service (ToS) field in the IP header is used. In this situation, the device copies the first three bits in the ToS field of the packet to the CoS (EXP) field in the MPLS header. The ToS value maps to one of the four priority queues on the device.

Configuring BGP or MPLS VPNs 36

Overview

PowerConnect B-MLXe supports the following BGP or MPLS VPN features:

- Defining a VRF Routing Instance
- MPLS Forwarding
- Generating Traps for VRFs
- Route Distinguisher to a VRF
- Automatic Route Filtering
- Assigning a VRF Routing Instance to a LAG Interface
- Cooperative Route Filtering
- Importing and Exporting Route Maps in a VRF
- Defining an External Community with a Route Map
- VPNv4 Route Reflector
- BGP VRF Load Sharing
- ECMP forwarding for IP VPN
- Autonomous System Number Override
- Allow Routes with its own AS Number
- Defining an External Community
- LSPs per VRF
- OSPF Sham Links
- OSPF on a PE Device to Redistribute BGP-VPNv4 Routes
- Adding a Static ARP Entry for a VRF
- Configuring an IP Static Interface Route Across VRFs
- IP TTL to MPLS TTL Propagation in an IPVPN
- Static Route within the VRF Context
- Backup Virtual Router for VRF Using VRRPE
- Ping and Traceroute for Layer-3 VPNs
- Displaying BGP or MPLS VPNv4 Information

This chapter describes how to configure BGP or MPLS VPNs on devices. BGP or MPLS VPNs as defined by RFC 2547 can be used by internet service providers to provide remote wide-area connectivity services using an MPLS Domain for data traffic and IBGP to distribute routing information. Each customer network can be completely segregated from every other customer network while sharing the same infrastructure.

This chapter is divided into the following sections:

- "What is a BGP or MPLS VPN" provides a basic description of BGP VPNs and an example. This section also lists the IETF RFCs and Internet Drafts supported by the implementation of BGP or MPLS VPNs.
- "BGP or MPLS VPN components and what they do" explains the software and hardware components that make up BGP or MPLS VPNs, including Customer Edge Router (CE), Provider Edge Routers (PE) and Virtual Routing and Forwarding (VRF) tables.
- "BGP or MPLS VPN operation" explains basic concepts about BGP or MPLS VPNs, including how routes are advertised and discovered, how the VRF manages packet forwarding, and how LSPs are employed to switch traffic across a MPLS Domain.
- "Configuring BGP VPNs on a PE" describes how to configure a BGP or MPLS VPN. In this section, the basic configuration steps of a BGP VPN are described.
- "Displaying BGP or MPLS VPNv4 information", "Displaying BGP or MPLS VRF information", and "Displaying additional BGP or MPLS VPN information" provide information about the display command you can use to manage a BGP or MPLS VPNv4 network.
- "BGP or MPLS VPN sample configurations" provide detailed examples of network configurations and examples of feature configurations.

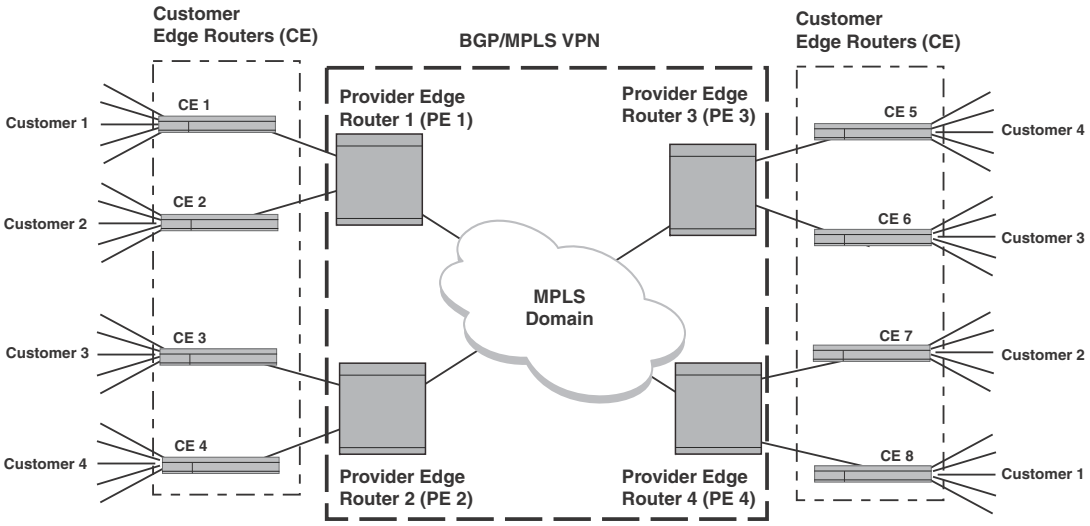
What is a BGP or MPLS VPN

MPLS provides scalable and efficient switching over an indeterminate group of devices along a predetermined Labeled Switch Path (LSP). Using MPLS, LSPs can be set statically or determined dynamically by the ISPs to provide traffic engineering features. BGP or MPLS VPNs build on this infrastructure to provide virtual-circuit connectionless service between remote sites. Using a common MPLS-domain, multiple Virtual Private Networks (VPNs) can be configured across a service-provider MPLS core network. Each VPN provides a secure data path that allows IP packetized traffic to share the infrastructure while being effectively segregated from other VPNs that are using the same MPLS Domain.

In Figure 202 four separate customers (1-4) each have remote sites. Each customer is connected to a network at a remote site through the MPLS domain while being completely segregated and secure from traffic between other sites. For instance, CE 1 and CE 8 belong to Customer 1. CE 1 is connected to the BGP or MPLS VPN network through PE 1 and CE 8 through PE 4. Using the service

provider's BGP or MPLS VPN service, traffic can be forwarded between CE1 and CE8 at the same time that Customers 2 through 4 use VPNs that operate over the same network infrastructure. Different customers can even use the same IP addresses without conflicting with other customers networks or creating any routing problems.

FIGURE 202 BGP or MPLS VPN network



IETF RFC and Internet Draft support

The implementation of BGP or MPLS VPNs supports the following IETF RFCs and Internet Drafts:

BGP or MPLS VPNs

RFC 4364: BGP or MPLS IP VPNs
RFC 4577: OSPF as the PE or CE Protocol in BGP or MPLS IP VPNs
RFC 4576: Using LSA Options Bit to Prevent Looping in BGP or MPLS IP VPNs (DN Bit)

BGP

RFC 1771 – A Border Gateway Protocol 4 (BGP-4)
RFC 1997 – BGP Communities Attribute
RFC 2283 – Multiprotocol Extensions for BGP-4
RFC 2842 – Capabilities Advertisement with BGP-4
RFC 2858 – Multiprotocol Extensions for BGP-4
RFC 3107 – Carrying Label Information in BGP-4

Draft standards

draft-ietf-idr-route-filter-11
draft-ietf-idr-bgp-ext-communities-07

MIB support

RFC 4382 – MPLS or BGP Layer 3 Virtual Private Network (VPN) Management Information Base.

BGP or MPLS VPN components and what they do

This section describes each of the following components, which make up a BGP or MPLS VPN (refer to Figure 203):

- Customer Edge device (CE)
- Provider Edge device (PE)
- Virtual Routing and Forwarding table (VRF)
- Provider MPLS Domain

The Customer Edge device (CE) – The CE provides connectivity with a customer's network and a Provider Edge device (PE). It can advertise routes available from the customer's network using RIP, OSPF or EBGP. Alternately, the CE can create a static default route to a PE. Outbound packets from a customer's network are forwarded from the CE to the PE, and inbound packets are forwarded from the PE to the CE attached to the customer's network.

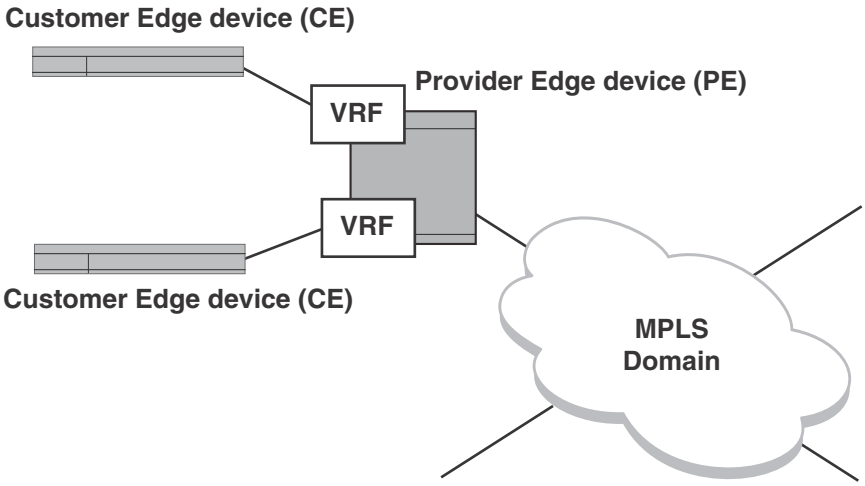
The Provider Edge device (PE) – In a BGP or MPLS VPN, the central component is the PE. The PE provides connectivity with the CE and with the MPLS Domain. On one side of the PE, routing information is exchanged with the CE using either static routes, RIP, OSPF, or EBGP. On the other side, IBGP is used with BGP multiprotocol extensions to communicate with all of the other PEs that

are connected to networks in the same VPN and available to the customer's network. When a CE sends packets to a PE to forward across an MPLS Domain, that PE functions as an MPLS ingress Label Edge device (LER) and the PE on the other end of the Domain functions as an MPLS egress LER.

Virtual Routing and Forwarding table (VRF) – The PE maintains a Virtual Routing and Forwarding table (VRF) for each customer that is attached to it through a CE. The VRF contains routes between the PE and the CE and Label Switched Paths (LSPs) across the MPLS domain for each PE that is a member of the customer's VPN. VRFs are defined on interfaces of the PEs.

Provider MPLS Domain – The Provider MPLS domain is composed of Provider (P) devices. An MPLS domain can traverse more than one service provider's MPLS network. The P devices do not store any VPN information; they just switch traffic from the ingress PE device along the LSP to the egress PE device.

FIGURE 203 BGP or MPLS VPN components



BGP or MPLS VPN operation

The purpose of a BGP or MPLS VPN is to forward packets between remote sites of a customer's network through a service provider's MPLS infrastructure. The section titled "BGP or MPLS VPN components and what they do" on page 1600 describes the network components required to perform that task. The following sections describe how those components work together to create this service:

- "Creating routes in a BGP or MPLS VPN"
- "Routing a packet through a BGP or MPLS VPN"

Creating routes in a BGP or MPLS VPN

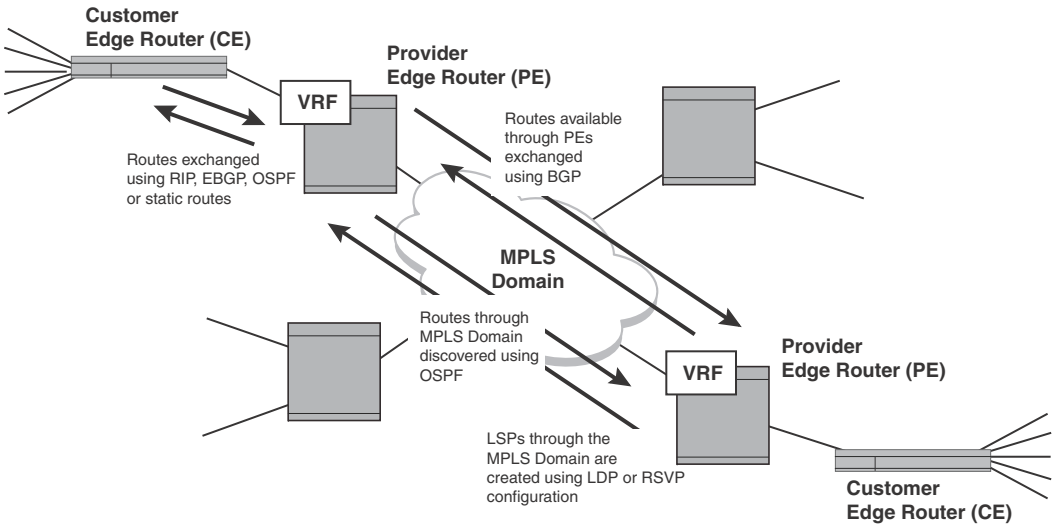
A CE device maintains the connection to the customer's network and is configured within that network to share access to its available network prefixes and to receive packets from other VPN-connected networks. That CE is connected to a PE through an interface that is configured for a specified VRF for connection to the BGP or MPLS VPN. This connection places the CE in the BGP or MPLS VPN. Routes that are available through the CE are then made available to the PE using RIP, OSPF, EBGP or a static route. These routes are then stored in the VRF where they are associated with the VPN. The route from the CE to the PE is kept in the CE's routing table.

The PE device is connected to the MPLS Domain through one or more interfaces. The PE must advertise the routes that it has available in it's VRF tables across the MPLS Domain to its PE peers. Available routes in the VRF are prepended with a Route Distinguisher (RD) and advertised across the MPLS Domain using IBGP. The PEs can either be configured for IBGP as either full mesh or with a route reflector to allow greater scalability. Routes that are advertised from other PEs in the VPN are received at the PE and collected in the VRF table. This procedure establishes which other PEs are in the VPN and what networks are available through them.

OSPF is used a the Interior Gateway Protocol (IGP) within the service provider's MPLS Domain to provide connectivity. OSPF also populates the traffic engineering (TE) used by RSVP-TE.

Labeled Switch Paths (LSPs) are then created using Label Distribution Protocol (LDP), Resource Reservation Protocol (RSVP) configurations in the MPLS Domain. Using this protocol, the PE obtains an LSP required to switch traffic to the other PEs. The network is now populated with all of the routes required to forward packets between the customer's networks.

FIGURE 204 BGP or MPLS VPN route discovery



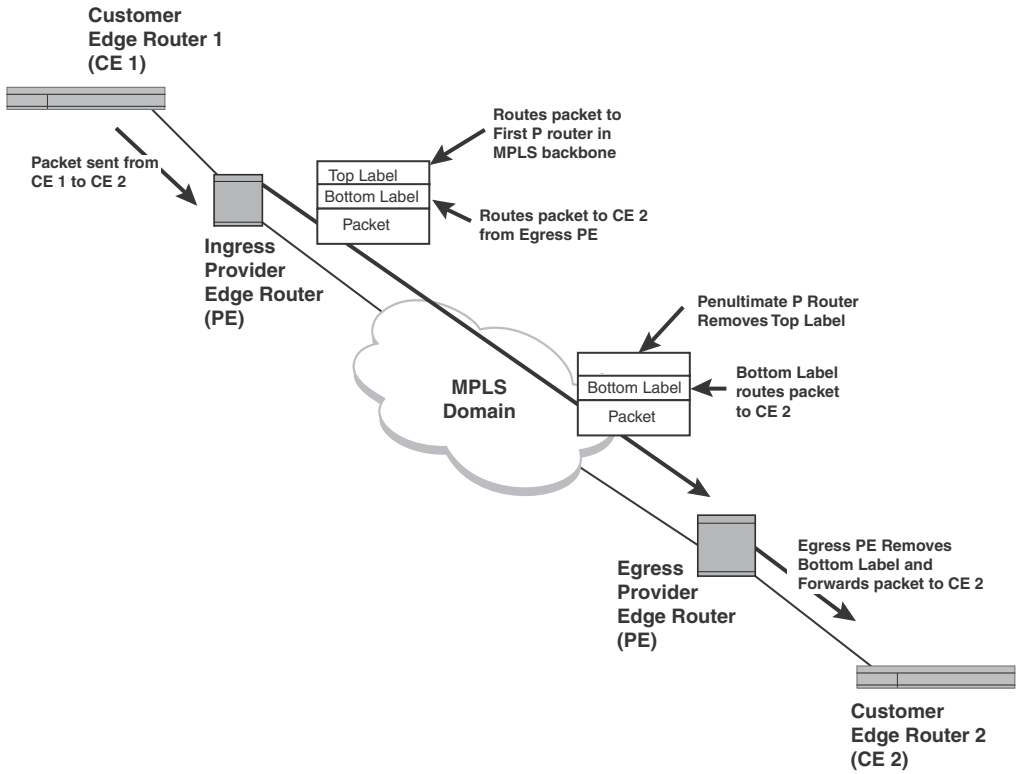
Routing a packet through a BGP or MPLS VPN

This section and the diagram in Figure 205 describe how a packet is forwarded through a BGP or MPLS VPN.

When a packet is forwarded from a CE to a PE, a bottom label is attached to the packet by the PE that is associated with the final destination. This label is obtained from the egress PE as part of the route discovery conducted by IBGP. Then, the top label which is obtained by the LSP connecting to the egress PE is added to the packet. The packet is then forwarded through the MPLS Domain and

is switched using the top label. At the penultimate device in the LSP, the top label is removed and the packet is forwarded to the egress PE. The egress PE uses the inner label to identify the CE to which the packet must be forwarded. The egress PE removes the inner label and forwards the packet to the correct CE.

FIGURE 205 Routing a packet through a BGP or MPLS VPN



Configuring BGP VPNs on a PE

To configure a BGP VPN on a Provider Edge device (PE) you must perform the configuration steps listed below.

1. "Defining a VRF routing instance"
2. "Configuring MPLS forwarding"
3. "Assigning a Route Distinguisher to a VRF"
4. 36, "Configuring BGP or MPLS VPNs"
5. "Defining automatic route filtering"
6. "Assigning a VRF routing instance to an interface"
7. "Assigning a VRF routing instance to a LAG interface"
8. "Setting up cooperative route filtering"
9. "Importing and exporting route maps"
10. "Defining an extended community for use with a route map"

11. “Creating a VPNv4 route reflector”
12. “Configuring BGP VRF load sharing”
13. “Configuring autonomous system number override”
14. “Configuring a PE to allow routes with its AS number”
15. “Setting up LSPs per VRF”
16. “Configuring OSPF sham links”
17. “Configuring OSPF on a PE device to redistribute BGP-VPNv4 routes”
18. “Generating traps for VRFs”

Defining a VRF routing instance

A single PE can contain one or more VRFs. Each of these VRFs must be defined separately on a PE. A PE will distribute routes and route packets to other members of the same VRF but not to other VRFs. The VRF name can be any string that you want to define it as.

To define the VRF routing instance VPN1 on a PE, enter the following command.

```
NetIron(config)# ip vrf VPN1
NetIron(config-vrf-vpn1)# exit-vrf
NetIron(config)#
```

Syntax: **[no] ip vrf** <*vrf_name*> **[max-routes** <*num*>**]**

Configures a VRF table on the device with the name **vrf_name** and puts the device in config-vrf mode.

The <*vrf_name*> parameter specifies a name for the VRF being created.

The **max-routes** parameter can be used to set the maximum number of routes (the <*num*> variable) that the VRF will accept. The default value for <*num*> is 5120. The acceptable range is from 128 to 262143.

Syntax: [no] exit-vrf

The **exit-vrf** command moves you out of the VRF configuration mode for the VRF you are configuring.

Configuring MPLS forwarding

A unique MPLS label is allocated for each VRF. This is called the per-VRF label. By default, this label is used as part of the BGP VPNv4 route when exchanging routes with a remote PE peer. When the per-VRF label is used, When the **label-switched** option in the **mpls-forwarding** command (within the IP VRF mode) is selected, a unique MPLS label is allocated for each BGP VPNv4 route. The following example enables mpls-forwarding for the VRF named VPN1.

```
NetIron(config)# ip vrf VPN1
NetIron(config-vrf-vpn1)# mpls-forwarding label-switched
NetIron(config-vrf-vpn1)# exit-vrf
NetIron(config)#
```

Syntax: [no] mpls-forwarding label-switched

When **mpls-forwarding** is set to **label-switched,** a unique MPLS label is allocated for each BGP VPNv4 route advertised to the remote PE peer. The forwarding behavior remains unchanged.

Assigning a Route Distinguisher to a VRF

Each instance of a VRF must have a unique Route Distinguisher (RD) assigned to it. The RD is prepended on any address being routed or advertised. The RD can be defined as either ASN-relative or IP address-relative. Since the RD is unique to an instance of a VRF, it allows the same IP address to be used in different VPNs without creating any conflict.

To assign a Route Distinguisher (RD) for a VRF based on the AS number 3 and the arbitrary identification number 6, enter the following command.

```
NetIron(config-vrf)# rd 3:6
```

Syntax: [no] rd <*route_distinguisher*>

The route_distinguisher variable specifies a route distinguisher for a VRF that gives a route associated with the VRF a unique identity. The RD is prepended on the address being advertised. The RD allows the same IP address to be used in different VPNs without creating any conflicts. It can also be used with the **route-target** command to constrain distribution of routes to or from a VPN. The route_distinguisher parameter can be either ASN-relative or IP address-relative as described:

ASN-relative – Composed of the local ASN number followed by a ":" and a unique arbitrary number. For example: 3:6.

IP address-relative – Composed of the local IP address followed by a ":" and a unique arbitrary number.

Defining IPv4 or IPv6 address families of a VRF

Each address family configuration level allows you to access commands that apply to that particular address family only.

To define IPv4 or IPv6 address families of a VRF, enter the following command.It will allow each address family to have its own max-route value.

```
NetIron(config)#vrf VPN1
NetIron(config-vrf-vpn1)#address-family ipv4 max-route 4544
NetIron(config-vrf-vpn1-ipv4)#exit-address-family
NetIron(config-vrf-vpn1)#exit-vrf
NetIron(config)#
```

Syntax: [**no**]**address-family**<ipv4/ipv6> [max-route <num>]

The max-route parameter can be used to set the maximum number of routes (the <*num*> variable) that the VRF will accept. The default value for <*num*> is 5120. The acceptable range is from 128 to 262143..

For IPv6 address family, the max-route acceptable range is 64 to 16384, with 128 as the default value.

Syntax: exit-address-family

The **exit-address-family** command moves you out of the ipv4 or ipv6 address family of a VRF you are configuring.

Defining automatic route filtering

Each VRF is configured with import and export route targets. The export route target sets an extended community attribute number that is appended to all routes that are exported from the VRF. The import route target value sets a filter that determines the routes that will be accepted into the VRF. Any route with a value in its import route-target contained in its extended attributes field matching the value in the VRF's import route target will be accepted. Otherwise the route will be rejected. This process is referred to as automatic route filtering.

To define an import route target of 3:6 and an export route target of 3:8 for a VPN, enter the following commands.

```
NetIron(config-vrf)# route-target import 3:6
NetIron(config-vrf)# route-target export 3:8
```

Syntax: [no] route-target [import | export | both] <*route-target*>

This command associates a route target specified by the route-target variable with a specified VRF for control on routes.

The **import** parameter specifies that routes with route-target extended community attributes matching the specified route-target variable can be imported into the VRF where this command is configured.

The **export** parameter specifies the route-target extended community attributes that are attached to routes export from the specified VRF.

The **both** parameter specifies that both the import and export values apply to the specified route-target variable for the VRF where this command is configured. This is the default state. It applies if no specific value for this parameter is set.

The <*route-target*> variable specifies a target VRF extended community. Like a route distinguisher, it is either AS-relative or IP address-relative.

Assigning a VRF routing instance to an interface

Once a VRF routing instance is defined, it must be assigned to one or more virtual or physical interfaces on a PE.

To assign the VRF named VPN1 to Ethernet interface 1/1, enter the following commands.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# ip vrf forwarding VPN1
```

Syntax: [no] ip vrf forwarding <*vrf-name*>

The <*vrf-name*> variable is the name of the VPN that the interface is being assigned to.

Assigning a VRF routing instance to a LAG interface

A VRF routing instance can be assigned to a dynamic LAG interface. To assign a VRF routing instance to a LAG the following rules must be observed:

- The dynamic LAG must be configured before assigning any of its ports to a non-default VRF routing instance.

- Before deployment of the dynamic LAG all members of the LAG must be in the default VRF routing instance.
- After the LAG is deployed, the primary port can be assigned to a non-default VRF routing instance.
- Once the dynamic LAG is deployed, all ports are in the LACP_BLOCK state until the LACP protocol can negotiate with the other end. Once the negotiation with the other end is completed, all the LAP ports are set to the FORWARD state.
- When the Dynamic LAG is undeployed, the primary port will stay in the VRF that it was assigned to but all secondary ports will move back to the default VRF.

The following configuration creates a dynamic LAG named “red” and assigns port 1/1 as the primary port and port 1/2 as a secondary port. The LAG is deployed and the primary port (1/1) is assigned to the VRF routing instance named “VPN1”. All ports in the LAG named “red” are then assigned to the VRF routing instance named “VPN1”.

```
NetIron(config)# lag red dynamic
NetIron(config-lag-red)# ports ethernet 1/1 to 1/2
NetIron(config-lag-red)# primary port 1/1
NetIron(config-lag-red)# ports ethernet 1/2
NetIron(config-lag-red)# deploy
NetIron(config-lag-red)# exit
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e10000-1/1)# ip vrf forwarding VPN1
```

If the dynamic LAG named “red” is undeployed as shown in the following, port 1/1 will remain in the VRF routing instance named “VPN1” but port 1/2 will be returned to the default VRF.

```
NetIron(config)# lag red dynamic
NetIron(config-lag-red)# no deploy
```

Setting up cooperative route filtering

Automatic route filtering in VRFs is provided through the **route-target** import command. By placing this command in the VRF configuration, routes can be filtered from being imported into a given VRF. Routes with extended community route targets matching the VRF’s import route-targets are permitted into a VRF. Otherwise, the routes are rejected.

The cooperative route filtering feature requires that you set a send command on the device that is sending the ORF, and a receive command on the device that is installing the ORF. To configure the sending device, use the following command in the VPNv4 address family.

```
NetIron(config-bgp-vpnv4u)# neighbor 3.3.3.1 capability orf extended-community
send-vrf-filter
```

Syntax: [no] neighbor <*neighbor_IPaddress*> **capability orf extended-community send-vrf-filter**

To configure the peering device use the following command in the VPNv4 address family.

```
NetIron(config-bgp-vpnv4u)# neighbor 3.3.3.2 capability orf extended-community
receive
```

Syntax: [no] neighbor <*neighbor_IPaddress*> **capability orf extended-community receive**

Importing and exporting route maps

Route-maps configured using the **route-map** command can be applied to a VRF to provide filtering of VPNv4 routes between PEs in a BGP or MPLS VPN. When a route-map is applied to a VRF, only VPNv4 routes are filtered. Other routes such as static routes, connected routes, OSPF VRF routes, or BGP CE side routes are not affected. Because the route map is applied to the VRF, it filters traffic to all connected PEs. This is in contrast to applying a route-map using the BGP neighbor. In that case, the route map applies to routes imported from or exported to the neighbor that is specified.

Route maps applied to a VRF can coexist with route maps that are applied to a BGP neighbor. You can filter routes from being imported into a VRF using the import and export route commands. This allows you to accept or deny the routes for one VRF without affecting the routes that are imported or exported from other VRFs. To do this, you must define a route-map import or export command.

To configure a VRF to apply the import route map ImportOne, use the following command at the VPNv4 prompt.

```
NetIron(config)# ip vrf vrfone
NetIron(config-ip-vrf-vrfone)# import map ImportOne
NetIron(config-ip-vrf-vrfone)# exit-vrf
NetIron(config)#
```

Syntax: [no] import map <*map-name*>

The map-name parameter is the name of the route map that you want to apply to the VRF.

To configure a VRF to apply the export route map ExportOne, use the following command at the VPNv4 prompt.

```
NetIron(config)# ip vrf vrfone
NetIron(config-ip-vrf-vrfone)# export map ExportOne
NetIron(config-ip-vrf-vrfone)# exit-vrf
NetIron(config)#
```

Syntax: [no] export map <*map-name*>

The **map-name** parameter is the name of the route map that you want to apply to the VRF.

Defining an extended community for use with a route map

Routes can be filtered in or out of a PE by the use of an IP extended community to identify them. In this situation, a route is identified by its extended community variable. It is entered as a route target in an IP extended community list and then matched in a route-map command. This route map is then applied from the PE that is defining the route to be filtered to the PE where the route filter is to be implemented by using a **neighbor** <*ip_address*> [**route-map**] command. If a VRF exists on the neighbor that exports the route-target being blocked, all routes from that VRF are blocked from being sent to the PE where the filter is defined.

To define the IP extended community list 20 to define route target RT 100:6 to be denied, enter the following command.

```
NetIron(config)# ip extcommunity-list 20 deny rt 100:6
```

Syntax: [no] ip extcommunity-list <*num*> **route-map [permit] [deny] [rt** <*route ID*>**] [soo** <*route ID*>**]**

The <*num*> variable is the extended community list number.

The **permit | deny** parameter indicates that action the device takes if the match is true.

The rt <*route ID*> variable specifies the route target that is applied to filtering. The <*route ID*> has the format of either ASN:nn or IP-address:nn. If four-byte ASNs have been enabled or if four-byte IP addresses are used, the user-purposed *nn* value can be a maximum of two bytes instead of our bytes.

The soo <*route ID*> variable specifies the site of origin. The <*route ID*> has the format of either ASN:nn or IP-address:nn. If four-byte ASNs have been enabled or if four-byte IP addresses are used, the user-purposed *nn* value can be a maximum of two bytes instead of our bytes.

Creating a VPNv4 route reflector

PE devices in a BGP or MPLS VPN share routes between each other using IBGP. This can be accomplished using a full mesh configuration or a route reflector can be used to simplify a networks topology and improve scalability. While the general concepts are the same for using Route Reflectors in a normal IBGP network as in an BGP or MPLS VPN, there are some differences. In addition, there are special conditions that apply when a route reflector is configured for normal IPv4 BGP traffic (IPv4) and for BGP or MPLS VPN traffic (VPNv4). The differences and special considerations are described in the following:

Special considerations when configuring a route reflector for both IPv4 and VPNv4:

- A VPNv4 route does not need to be installed in any VRF before being reflected.
- Route reflector configurations for IPv4 and VPNv4 are separated in different address family configurations.
- For a VPNv4 route installed to a VRF, the reflected VPNv4 route still carries the original RD and PA.
- If there is a route reflector configuration change, a warning message is displayed that requests the user to clear the neighbor session.

Specific commands for VPNv4 – There are VPNv4 specific commands that must be configured to configure a route reflector for a BGP or MPLS VPN under address family VPNv4. A route reflector can be configured on a PE for IPv4 and VPNv4 or for either exclusively. If you are configuring a route reflector for a BGP or MPLS VPN, you must configure it specifically using the VPNv4 specific commands.

To create a VPNv4 route reflector with a client at the IP address 11.11.11.2, enter the following commands at the vpnv4 level of BGP Config level.

```
NetIron(config-bgp-vpnv4u)# neighbor 11.11.11.2 route-reflector-client
```

Syntax: [no] neighbor <*IPaddress*> **route-reflector-client**

The <*IPaddress*> variable is the IP address of the PE device that you want to define the route reflector client.

A route reflector can be setup with local import filtering to filter out VPNv4 routes matched by an extended community list. This requires that you create an extended community list for the routes you want to filter and set the following command.

```
NetIron(config-bgp)# address-family vpnv4 unicast
NetIron(config-bgp-vpnv4u)# rr-group 1
```

Syntax: [no] rr-group <*group-num*>

The <*group-num*> variable refers to an extended community list number from 1 to 99 that specifies the routes that you want to filter.

Configuring BGP VRF load sharing

The default for each VRF is to maintain only the lowest-cost route in its routing table for each VPN that it is connected to. If a lower-cost route is discovered, it will replace the route that is currently in the table. If another route of equal cost is discovered, it will be rejected. The PowerConnect however is able to perform load sharing over multiple routes to the same destination. In order to make this feature operational, you must increase the number of path entries allowed in a VRF's routing table.

Configuring BGP VRF load sharing requires two different CLI commands that work in relationship with each other. These are the global **ip load-sharing** command and the BGP VRF specific **maximum-path** command. The value set for ip load-sharing provides a maximum number that the maximum-path value for a specific route can be set to. The **maximum-path** command has a maximum value of 8. If the ip load-sharing value is set to 4 or greater, the maximum-path value for a specific BGP VRF can be set to a value of from 1 to 4. The default is 1. If the ip load-sharing value is set to less than 4, the maximum-path value for a specific BGP VRF can only be set to the global ip load-sharing value or less.

To set the **maximum-path** value to 4, enter the following commands at the VPNv4 level of the BGP configuration level.

```
NetIron(config-bgp)# maximum-paths 4
```

Syntax: [no] maximum-paths <*num*>

The <*num*> variable is the maximum number of routes that can be maintained for a VRF. The default value is 1. The maximum value is 4. The value cannot exceed the value set for the device by the **ip load-sharing** command.

ECMP forwarding for IP VPN

ECMP hardware forwarding is now supported for IP VPN packets when an outgoing interface is configured as a physical port and a VE interface. If multiple routes are using ECMP to route to a destination, then hardware ECMP is automatically enabled. ECMP load sharing for IP over MPLS is supported for 2-8 tunnels. For more information on configuring ECMP load sharing for IP VPN, refer to "Configuring BGP VRF load sharing" on page 1610.

ECMP hardware forwarding is supported only in static CAM mode. ECMP hardware forwarding is not supported for dynamic mode. Hitless upgrade for ECMP hardware forwarding is not supported.

When configuring ECMP hardware forwarding, all outgoing paths must configured in the same VRF. A hash value is computed for a packet when it is received by XPP. XPP will use the hash value to select a PRAM that is used to forward the packet to its destination. An ECMP PRAM block consists of 8 PRAMs. The hash value for each outgoing packet on a Customer Edge device interface is calculated based on the source MAC address, destination MAC address, VLAN ID, source IP address, destination IP address, IP protocol field, TCP or UDP source port, and TCP or UDP destination port.

The hash value for each incoming packet on the route target is calculated based on the source MAC address, destination MAC address, VLAN ID, source IP address, destination IP address, IP protocol field, TCP or UDP source port, TCP or UDP destination port, and a VC label for an MPLS packet.

Configuring autonomous system number override

There are some situations where a customer will want to connect to a service provider's BGP or MPLS VPN network using the same AS number at more than one site. This can create a problem because it is the default BGP procedure to reject routes from the same AS. One solution to this problem is to configure a PE router to override the AS_PATH attribute of its BGP neighbor. This is accomplished by configuring the **neighbor** <*ip_address*> **as-override** command on the PE. When this is enabled, the PE device determines when the AS_PATH attribute in a route intended for a neighbor CE contains the same AS number as the CE. When this is determined, the PE device substitutes its own AS number for the CE's in the AS_PATH attribute. The CE is then able to receive the route. The following additional conditions apply when this feature is in effect:

- In a situation where the AS_PATH attribute contains more than one occurrence of the CE's AS number in the initial sequence, the PE device will replace all those occurrences with its own AS number.
- The PE device will add its own AS number to the AS_PATH attribute just as it would normally.

The following command configures the PE device to replace its attached CE's AS number with its own AS number. BGP neighbor at IP address 33.33.36.2 the configuration of PE 2 required to enable Autonomous System number override for the BGP neighbor CE 2.

To configure a PE device to replace its attached CE's AS number with its own AS number, enter the following commands at the VRF level of the BGP Config level.

```
NetIron(config-bgp-vpnv4u)# neighbor 33.33.36.2 as-override
```

Syntax: [no] neighbor <*IPaddress*> **as-override**

The <*IPaddress*> variable is the IP address of the CE whose AS number is being replaced with the PE's AS number.

Configuring a PE to allow routes with its AS number

BGP rejects routes that contain its own AS number within its AS_PATH attribute to prevent routing loops. In an MPLS or VPN hub and spoke topology this can stop legitimate routes from being accepted. The **allowas-in** command fixes this problem by allowing you to set a parameter that disables the AS_PATH check function for routes learned from a specified location.

To configure a PE to disable the AS_PATH check function for routes sent to it by its BGP neighbor (a CE device with the IP address 33.33.36.2) for a maximum limit of 3 occurrences of the route, enter the following command at the BGP VRF configuration level.

```
NetIron(config-bgp-ipv4u-vrf)# neighbor 33.33.36.2 allowas-in 3
```

Syntax: [no] neighbor <*IPaddress*> **allowas-in** <*asn_limit*>

The <*IPaddress*> is the IP address of the neighbor CE device from which the PE device can accept routes that have the same AS number.

The asn_limit value prevents loops by limiting the number of occurrences that the PE's AS number can be accepted in routes that are received from the specified device.

Setting up LSPs per VRF

IBGP is used between PEs to determine routes that are available between VRFs. These routes are linked to a Label Switched Path (LSP) that has been defined separately either as a static path or using LDP or RSVP. The LSP is used to tunnel through the MPLS Domain to the destination PE. Under most circumstances, the default route between two PEs will be chosen by IBGP between the VRFs with the PE's loopback address as the next hop. If there is a single loopback on the PE, the same LSP tunnel will be the only path used between any VRF defined on a PE and VRFs on other specified PEs.

More than one LSP can be configured between PEs however, where each LSP is associated with a different Loopback address on the PE. In this case, any loopback address on a PE can be assigned as the nexthop address for a specific or multiple VRFs. This allows you to assign some VRFs on a PE to one LSP and other VRFs to a different LSP. Through this method, traffic from different VRFs can be assigned to LSPs that provide different qualities of service. This feature can also be employed to provide for load-balancing across the MPLS domain.

To configure a PE device to use different LSPs, a BGP next hop must be configured for a VRF as the following example illustrates.

```
NetIron(config)# ip vrf blue
NetIron(config-ip-vrf-blue)# bgp next-hop loopback 2
NetIron(config-ip-vrf-blue)# exit-vrf
NetIron(config)#
```

Syntax: [no] bgp next-hop <*loopback-interface*>

The <*loopback-interface*> variable is the number of the loopback interface that you will be assigning to the VRF as a BGP next hop. The loopback address becomes the defined VRF's nexthop for its VPNv4 routes that are sourced by this device only when:

- The loopback interface exists and has an IP address set.
- The loopback interface has an IP a subnet mask of /32
- The loopback interface is in the default VRF.

If these conditions are not met, the default nexthop is used.

For a detailed example of this feature refer to "Setting an LSP for each VRF on a PE" on page 1709

Configuring OSPF sham links

OSPF can be used to propagate links between a Customer Edge device (CE) and a Provider Edge device (PE). Normal operation of this type of network assumes that the only connections between CEs pass through the provider network. However, if other links or routes between the CEs exist within the same area, problems can arise due to the OSPF preference for Intra-area links over Inter-area links.

Problems can be avoided by creating a virtual intra-area OSPF link between two PEs. This virtual link is called a sham link. If the OSPF instances exist in the same area, a sham link causes OSPF to treat the route through the service provider network as an intra-area link instead of an inter-area link.

NOTE
If no backdoor link exists, no purpose exists for creating a sham link.

A cost is assigned to the sham link to help the OSPF network determine when to route over the sham link and when to route over the backdoor link. Because this virtual link (sham-link) appears as an intra-area link, the OSPF areas in which each of the PEs reside must be the same.

To configure an OSPF sham link, use the command for creating a sham link on both the local device and the remote PE device. Before attempting to create a sham link, note the following important information:

- For sham links to work, OSPF cannot be configured on the loopback interface in the applicable area.
- The redistribution of BGP to OSPF must be configured.
- A BGP VPN4 route to the loopback address must exist in both of the pertinent VRFs' routing tables.
- After the BGP VPN4 route exists in the VRF IP route table, the hello (and other) packet exchanges can go through for sham links even if the backdoor CE link does not exist.

The first example that follows illustrates the command for creating an OSPF sham link between PE devices. The command shows the command entry on one device with a source IP address of 2.2.2.1 and destination address of 2.2.2.2. The second example shows the complete configuration sequence (from both PE devices) and shows the command for viewing the sham link. (Refer to "Displaying OSPF sham links" on page 1681 for the display contents.)

Use this command in the OSPF VRF configuration level.

```
NetIron(config-ospf-router)# area 1 sham-link 2.2.2.1 2.2.2.2 cost 10
```

Syntax: **[no] area** <*area_id*> **sham-link** <*source_address*> <*destination_address*> **cost** <*cost_value*>

Possible values:

The area_id variable is the ID number of the OSPF area assigned to the sham ink being defined in this command.

The source_address variable is the IP address of the source PE device.

The destination_address variable is the IP address of the destination PE device.

The cost_value variable sets the OSPF cost for sending packets over the sham link. This parameter can be a numeric value in the range 1 – 65535.

The following illustrates the configuration that takes place first on PE1 and then on PE2:

Sham link configuration on PE1

```
router ospf vrf CustomerA
area 1
area 1 sham-link 172.31.255.1 172.31.255.2 cost 1
redistribution bgp

interface loopback 2
ip vrf forwarding CustomerA
ip address 172.31.255.1/32
!
```

```
NetIron@Router1#sh ip route vrf CustomerA 172.31.255.2
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
Destination Gateway Port Cost Type Uptime
1 172.31.255.2/32 172.30.255.48 lsp PE1-PE2 200/0 B 10m3s
```

Sham link configuration on PE2

```
router ospf vrf CustomerA
area 1
area 1 sham-link 172.31.255.2 172.31.255.1 cost 1
redistribution bgp

interface loopback 2
ip vrf forwarding CustomerA
ip address 172.31.255.2/32
!
NetIron@Router2#show ip route vrf CustomerA 172.31.255.1

Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
Destination Gateway Port Cost Type
1 172.31.255.1/32 172.30.255.32 lsp PE2-PE1 200/0 B
```

Configuring OSPF on a PE device to redistribute BGP-VPNv4 routes

To allow OSPF route exchange between a specified VRF on a PE device and its associated CE device, OSPF must be configured to redistribute BGP routes from the local AS as described in the following steps:

- "Defining an OSPF instance in a VRF"
- "Creating an OSPF area in an OSPF VRF instance"
- "Creating a domain identifier in an OSPF VRF instance"
- "Assigning a domain tag in an OSPF VRF instance"

Defining an OSPF instance in a VRF

To define an OSPF instance in VRF VPN1, enter the following command at the OSPF Config level.

```
NetIron(config)# router ospf vrf VPN1
```

Syntax: [no] router ospf vrf <*vrf_name*>

The <*vrf_name*> value specifies the name of the VRF that you are creating an instance of OSPF in.

Creating an OSPF area in an OSPF VRF instance

To create OSPF area 1 in OSPF VRF instance VPN1, enter the following command in the OSPF VRF Instance Config level.

```
NetIron(config-ospf-router)# area 1
```

Syntax: [no] area <*area_id*>

The <*area-id*> value is the number of the OSPF area instance being created.

Creating a domain identifier in an OSPF VRF instance

To create OSPF domain identifier 0.0.0.100 in OSPF VRF instance VPN1, enter the following command in the OSPF VRF Instance Config level.

```
NetIron(config-ospf-router)# domain-id 0.0.0.100
```

Syntax: [no] domain-id <*domain_identifier*>

The <*domain_identifier*> value specifies an four-byte quantity.

Assigning a domain tag in an OSPF VRF instance

To assign OSPF domain tag 1200 in OSPF VRF instance VPN1, enter the following command in the OSPF VRF Instance Config level.

```
NetIron(config-ospf-router)# domain-tag 1200
```

Syntax: [no] domain-tag <*domain_tag*>

The <*domain_tag*> parameter specifies an arbitrary four-byte quantity. It is added in tag fields of Type-5 and Type-7 LSAs generated by a PE device for redistributed BGP-VPNv4 routes.

If not specified, the domain-tag value is calculated from the autonomous system number of the MPLS Domain.

Adding a static ARP entry for a VRF

To configure a static ARP entry to a VRF enter the following command at the global configuration level.

```
NetIron(config)# arp vrf green <num> <ip-addr> <mac-addr> ethernet <portnum>
```

Syntax: [no] arp vrf <*vrf-name*> <*num*> <*ip-addr*> <*mac-addr*> **ethernet** <*portnum*>

The <*vrf-name*> parameter specifies the VRF you are configuring a static ARP entry for.

The <*num*> parameter specifies the entry number. You can specify a number from 1 up to the maximum number of static entries allowed on the device. You can allocate more memory to increase this amount. To do so, enter the system-max ip-static-arp <*num*> command at the global CONFIG level of the CLI.

NOTE
This has be deprecated as of release 3.6.

The <*ip-addr*> command specifies the IP address of the device that has the MAC address of the entry.

The <*mac-addr*> parameter specifies the MAC address of the entry.

The ethernet <*portnum*> command specifies the port number attached to the device that has the MAC address of the entry.

To clear the ARP entries for a specified VRF, enter the following command.

```
NetIron# clear arp vrf blue
```

Syntax: clear arp vrf <*vrf-name*>

The <*vrf-name*> parameter specifies the VRF you want to clear all ARP entries for.

Configuring IP TTL to MPLS TTL propagation in an IPVPN

The **vrf-propagate-ttl** and **label-propagate-ttl** commands have been added which configure the device to propagate TTL values in an IPVPN between the IP TTL value and the MPLS TTL value as described in Table 272.

TABLE 272 MPLS TTL propagation behavior with IPVPNs on the PowerConnect B-MLXe.

With vrf-propagate-ttl and label-propagate-ttl configured	Without vrf-propagate-ttl and label-propagate-ttl configured (default)
• At the ingress device, the IP TTL value -1 is copied to both the tunnel label and the VC label. • At the transit device, the tunnel label is decremented by 1. • At the PHP device the Tunnel label TTL is set to the VC label and the tunnel label is popped. • At the egress device, the IP TTL value is set to min (VC label TTL, IP TTL) and the VC label is popped. The IP TTL value is then decremented by 1 if it is being forwarded out of the device.	• At the ingress device, both the tunnel TTL value and the VC label TTL value are set to 255 • At the transit device, the tunnel label is decremented by 1. • At the PHP device the Tunnel label TTL is popped without changing the VC label's TTL. • At the egress device, the VC label is popped without copying the TTL value to the IP packet. The IP TTL value is then decremented by 1 if it is being forwarded out of the device.

To configure a PowerConnect B-MLXe to propagate the IP TTL values to and from the MPLS TTL values in an IPVPN, enter **vrf-propagate-ttl** and **label-propagate-ttl** commands as shown in the following.

```
NetIron(config)# router mpls
NetIron#(config-mpls)# policy
NetIron#(config-mpls-policy)# vrf-propagate-ttl
NetIron#(config-mpls-policy)# label-propagate-ttl
```

To return the condition to the default off state if the **label-propagate-ttl** command has been previously configured, enter a command such as the following.

```
NetIron(config)# router mpls
NetIron#(config-mpls)# policy
NetIron#(config-mpls-policy)# no vrf-propagate-ttl
```

Syntax: [no] vrf-propagate-ttl

Using the **no** option returns the condition to the default off state if the **vrf-propagate-ttl** command has been previously configured.

Syntax: [no] label-propagate-ttl

Using the **no** option returns the condition to the default off state if the **label-propagate-ttl** command has been previously configured.

Configuring a static route within the VRF context

To configure a static route entry in a VRF, enter the following command.

```
NetIron#(config) ip vrf blue
NetIron#(config-vrf-blue) ip route 192.0.0.0 255.0.0.0 195.1.1.1
```

Syntax: [no] ip route <*dest-ip-addr*>/<*mask-bits*> <*next-hop-ip-addr*> [<*metric*>]

The <*dest-ip-addr*> is the route's destination. The <*dest-mask*> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24. To configure a default route, enter 0.0.0.0 for <*dest-ip-addr*> and 0.0.0.0 for <*dest-mask*> (or 0 for the <*mask-bits*> if you specify the address in CIDR format). Specify the IP address of the default gateway using the <*next-hop-ipaddr*> parameter.

The <*next-hop-ip-addr*> is the IP address of the next-hop device (gateway) for the route.

The <*metric*> parameter specifies the cost of the route and can be a number from 1 – 16. The default is 1.

NOTE
Note that the **ip route** command is executed at the "config-vrf" configuration level.

Configuring an IP Static interface route across VRFs

You can configure an IP Static interface route from one VRF to an IP interface in a different VRF. This is described in "Configuring an IP Static interface route across VRFs" on page 1617.

Configuring a backup Virtual Router for VRF using VRRPE

With this release, you can use the Virtual Router Redundancy Protocol Extended (VRRPE) to provide a redundant connection to a VRF instance in a BGP or MPLS VPN. This is accomplished by assigning the VRRPE interface to a port that is also assigned to the VRF.

Configuration of VRRPE support for a VRF must be accomplished in the order described below.

1. Enable an interface
2. Enable VRF forwarding on the interface
3. Configure an IP address on the interface
4. Enable VRRPE on the interface and set the VRID
5. Configure the IP address for the Virtual Router
6. Activate the virtual interface

DANGER
You must configure a VRF on an interface before configuring a Virtual Router (VRRPE) on it. If you enable the Virtual Router before you enable the VRF, the Virtual Router configuration will be deleted.

Configuration example

The following example configures a backup virtual device using VRRPE for VRF "blue" on an Ethernet interface.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-3/1)# ip vrf forwarding blue
NetIron(config-if-3/1)# ip address 1.2.3.1/8
NetIron(config-if-3/1)# ip vrrp-extended vrid 1
NetIron(config-if-3/1-vrid-1)# backup
NetIron(config-if-3/1-vrid-1)# ip-address 1.2.3.10
NetIron(config-if-3/1-vrid-1)# activate
```

Ping and Traceroute for layer-3 VPNs

The Ping and Traceroute utilities have been enhanced in release 02.1.00 to help with management of Layer-3 VPNs:

Ping VRF

A VRF option has been added to the **ping** command. To use this option, enter the following command.

```
PE1# ping vrf blue 10.10.10.10
```

Syntax: ping vrf <*vrf-name*> <*ip-address*>

The <*vrf-name*> is the name of the VRF that you want to send a ping packet to.

The <*ip-address*> is the ip address containing the VRF tp which you want to send a ping packet .

Traceroute VRF

A VRF option has been added to the **traceroute** command. To use this option, enter the following command.

```
PE1# traceroute vrf blue 10.10.10.10
```

Syntax: traceroute vrf <*vrf-name*> <*ip-address*>

The <*vrf-name*> is the name of the VRF that you want to conduct a traceroute to.

The <*ip-address*> is the ip address containing the VRF to which you want to conduct a traceroute.

Generating traps for VRFs

You can enable and disable SNMP traps for VRFs. VRF traps are enabled by default.

To enable VRF traps after they have been disabled, enter the following command.

```
NetIron(config)# snmp-server enable traps vrf
```

Syntax: [no] snmp-server enable traps vrf

Use the **no** form of the command to disable VRF traps.

Displaying BGP or MPLS VPNv4 information

You can display the following information about a BGP or MPLS VPN configuration on the device:

- "Displaying VPNv4 route information"
- "Displaying VPNv4 route information for a specified IP address"
- "Displaying VPNv4 attribute entries information"
- "Displaying VPNv4 dampened paths information"
- "Displaying VPNv4 filtered routes information"
- "Displaying VPNv4 Flap statistics information"
- "Displaying VPNv4 route distinguisher information"
- "Displaying VPNv4 neighbor information"
- "Displaying advertised routes for a specified VPNv4 neighbor"
- "Displaying attribute entries for a specified VPNv4 neighbor"
- "Displaying Flap statistics for a specified VPNv4 neighbor by IP address"
- "Displaying received ORFs information for a specified VPNv4 neighbor"
- "Displaying a specified neighbor VPNv4 routes"
- "Displaying routes summary for a specified VPNv4 neighbor"
- "Displaying summary route information"
- "Displaying the VPNv4 route table"
- "Displaying BGP VPNv4 MPLS tag information"

Displaying VPNv4 route information

You can display route information about VPNv4 routes by entering the following command at any level of the CLI.

```
NetIron# show ip bgp vpnv4
Total number of BGP VPNv4 Routes: 285
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
Route Distinguisher: 1:1
*i  10.80.1.1/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.2/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.3/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.4/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.5/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.6/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.7/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.8/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.9/32       2.2.2.2         100    0      206 311 i
*i  10.80.1.10/32      2.2.2.2         100    0      206 311 i
*i  10.80.1.11/32      2.2.2.2         100    0      206 311 i
*i  10.80.1.12/32      2.2.2.2         100    0      206 311 i
*i  10.80.1.13/32      2.2.2.2         100    0      206 311 i
*i  10.80.1.14/32      2.2.2.2         100    0      206 311 i
*i  10.80.1.15/32      2.2.2.2         100    0      206 311 i
*i  10.80.1.16/32      2.2.2.2         100    0      206 311 i
*i  10.80.1.17/32      2.2.2.2         100    0      206 311 i
*i  10.80.1.18/32      2.2.2.2         100    0      206 311 i
--More--, next page: Space, next line: Return key, quit: Control-c
```

This display shows the following information.

TABLE 273 BGP4 summary information

This field...	Displays...
Total number of BGP VPNv4 Routes:	The number of BGP VPNv4 routes.
Status or Status Codes	The route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE:** If the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the received better routes from other sources (such as OSPF, RIP, or static IP routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – INTERNAL. The route was learned through BGP4. • L – LOCAL. The route originated on this device. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **NOTE:** This field appears only if you enter the **route** option.
Origin code	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.
Route Distinguisher	A unique ID that is prepended on any address being routed or advertised from a VRF. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" and a unique arbitrary number. For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" and a unique arbitrary number.
Network	IP address or mask of the destination network of the route.
Next Hop	The next-hop device for reaching the network from this device.
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares route on the basis of local preference, the route with the higher local preference is chosen. The preference can have a value from 0-4294967295

TABLE 273 BGP4 summary information (Continued)

This field...	Displays...
Weight	The value that this route associates with routes from a specific neighbor. For example, if the device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Path	The routes AS path.

To clear the VPNv4 routing table, you must enter the following commands.

```
NetIron# clear ip bgp vpnv4 neighbor all soft out
NetIron# clear ip bgp vpnv4 neighbor all soft in
```

Syntax: clear ip bgp vpnv4

The **dampening** parameter clears route flap dampening information.

The **flap-statistics** parameter clears route flap statistics.

The **neighbor** parameter clears BGP neighbors.

Displaying VPNv4 route information for a specified IP address

To display only the routes to a specified network, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vpnv4 10.2.2.0/24
Route Distinguisher: 2:1
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network          Next Hop      Metric LocPrf Weight Path
*i  10.2.2.0/24        4.4.4.4        1      100     0       ?
```

Syntax: show ip bgp vpnv4 <*ip-addr*>

The <*address*> <*mask*> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

The **Number of BGP Routes** matching display conditions field in this display is described in Table 274 below. For information about all other fields in this display, refer to Table 273.

TABLE 274 Route flap dampening statistics

This field...	Displays...
Number of BGP Routes matching display conditions	The number of routes to the network specified as a parameter in the **show ip bgp vpnv4** <*ip-addr*> command.

Displaying VPNv4 attribute entries information

The route-attribute entries table lists the sets of BGP VPNv4 attributes stored in the device memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the router typically has fewer route attribute entries than routes. To display the route-attribute entries table at any level of the CLI.

```
NetIron# show ip bgp vpn attribute-entries
        Total number of BGP Attribute Entries: 55
1       Next Hop  :0.0.0.0            Metric   :0               Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 600:1
        AS Path   :310
        Address: 0x24644060  Hash:45 (0x0100036e) Reference Counts: 0:0:30
2       Next Hop  :0.0.0.0            Metric   :0               Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 600:1
        AS Path   :311
        Address: 0x24645f48  Hash:47 (0x01000370) Reference Counts: 0:0:30
3       Next Hop  :2.2.2.2            Metric   :0               Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 100:1 RT 200:1
        AS Path   :206 311
        Address: 0x24645538  Hash:276 (0x0100087a) Reference Counts: 30:0:0
```

This display shows the following information.

TABLE 275 BGP VPNv4 attribute entries

This field...	Displays...
Total number of BGP Attribute Entries	The number of routes contained in the BGP4 route table for this device.
Next Hop	The IP address of the next hop device for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.

TABLE 275 BGP VPNv4 attribute entries (Continued)

This field...	Displays...
Aggregator	Aggregator information: • **AS Number** shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. **Router-ID** shows the device that originated this aggregator
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss. • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
Extended Community	The extended community attributes.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	This is an internal value used for debugging purposes only.
Hash	This is an internal value used for debugging purposes only.
Reference Counts	This is an internal value used for debugging purposes only.

Displaying VPNv4 dampened paths information

To view BGP VPNv4 paths suppressed due to dampening, enter the following command.

```
NetIron# show ip bgp vpnv4 dampened-paths
```

Displaying VPNv4 filtered routes information

To view BGP VPNv4 filtered paths information, enter the following command.

```
NetIron# show ip bgp vpnv4 filtered-routes
```

Displaying VPNv4 Flap statistics information

To display route flap statistics for all routes, enter the following command at any level of the CLI.

```
NetIron# show ip bgp vpnv4 flap-statistics ?
```

Syntax: **show ip bgp vpnv4 flap-statistics [regular-expression** <*regular-expression*> **|** <*address*> <*mask*>
[longer-prefixes] | neighbor <*ip-addr*> **| filter-list** <*num*>**...]**

The <*address*> <*mask*> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

The **as-path-filter** <*num*> parameter specifies one or more filters. Only the routes that have been dampened and that match the specified filter or filters are displayed.

The **neighbor** <*ip-addr*> parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbor** <*ip-addr*> **flap-statistics**.

The **regular-expression** <*regular-expression*> parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters.

This display shows the following information.

TABLE 276 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4 route table that have changed state and thus have been marked as flapping routes.

You also can display all the dampened routes by entering the following command: **show ip bgp dampened-paths**.

Displaying VPNv4 route distinguisher information

In order to view the BGP VPNv4 information for routes that contain a specific route distinguisher, enter the following command.

```
NetIron# show ip bgp vpnv4 rd 5:1 detail
Total number of BGP Routes: 34
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
1       Prefix: 10.6.1.0/24,  Status: I,  Age: 16h9m21s
         NEXT_HOP: 4.4.4.4,  Learned from Peer: 4.4.4.4 (1)
         Out-Label: 500000
          LOCAL_PREF: 100,  MED: 2,  ORIGIN: incomplete,  Weight: 0
           AS_PATH:
            Extended Community: RT 300:1 RT 100:2 RT 100:3
2       Prefix: 10.40.1.1/32,  Status: I,  Age: 16h9m21s
        NEXT_HOP: 4.4.4.4,  Learned from Peer: 4.4.4.4 (1)
         Out-Label: 500000
          LOCAL_PREF: 100,  MED: 2,  ORIGIN: incomplete,  Weight: 0
           AS_PATH:
            Extended Community: RT 300:1 RT 100:2 RT 100:3
3       Prefix: 10.40.1.2/32,  Status: I,  Age: 16h9m21s
         NEXT_HOP: 4.4.4.4,  Learned from Peer: 4.4.4.4 (1)
         Out-Label: 500000
          LOCAL_PREF: 100,  MED: 2,  ORIGIN: incomplete,  Weight: 0
           AS_PATH:
            Extended Community: RT 300:1 RT 100:2 RT 100:3
```

TABLE 277 BGP VPNv4 route distinguisher entries

This field...	Displays...
Total number of BGP Routes	The number of routes contained in the BGP4 route table that contain the specified RD.
Prefix	The network address and prefix.
Age	The last time an update occurred.
Learned from Peer	The IP address of the neighbor that sent this route.
Out-Label	MPLS label associated with this device.
MED	The route's metric. If the route does not have a metric, this field is blank.
AS Path	The route's AS path.
Extended Community	Extended community attributes associated with this device.

Displaying VPNv4 neighbor information

To view BGP4 configuration information and statistics for VPNv4 neighbors, enter the following command.

```
NetIron# show ip bgp vpnv4 neighbors
    Total number of BGP Neighbors: 2
1   IP Address: 2.2.2.2, AS: 1 (IBGP), RouterID: 2.2.2.2, VRF: default
    State: ESTABLISHED, Time: 14h47m39s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 21 seconds, HoldTimer Expire in 141 seconds
       UpdateSource: Loopback 1
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       40      887       0            0
       Received: 1       35      887       0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  unicast  capability
      Peer Negotiated VPNv4 unicast  capability
      Peer configured for IPV4 unicast  Routes
      Peer configured for VPNv4 unicast  Routes
    TCP Connection state: ESTABLISHED
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:   29202, Received: 28108
       Local host:  3.3.3.3, Local  Port: 179
       Remote host: 2.2.2.2, Remote Port: 8079
       ISentSeq:    7683960  SendNext:    7713163  TotUnAck:          0
       TotSent:       29203  ReTrans:           0  UnAckSeq:    7713163
       IRcvSeq:   256457831  RcvNext:   256485940  SendWnd:       65000
       TotalRcv:      28109  DupliRcv:          0  RcvWnd:        65000
       SendQue:           0  RcvQue:            0  CngstWnd:       1479
```

This example shows how to display information for VPNv4 neighbors. None of the other display options are used; thus, all of the information is displayed for all neighbors. The number in the far left column indicates the neighbor for which information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in theTransmission Control Block (TCB) for the TCP session between the device and a neighbor. These fields are described in detail in section *3.2 of RFC 793*, "*Transmission Control Protocol Functional Specification*".

Syntax: **show ip bgp vpnv4 neighbors** [*<ip-addr>* **[advertised-routes [detail** [*<ip-addr>*[*/<mask-bits>*]]]] **|**
[attribute-entries [detail]] | [flap-statistics] | [last-packet-with-error] | [received extended-community] | [received prefix-filter] | [routes [best] | [detail [best] | [not-installed-best] | [unreachable]] |
[rib-out-routes [*<ip-addr>*/*<mask-bits>* **|** *<ip-addr> <net-mask>* **| detail]] |**
[routes-summary]]

The *<vrf-name>* parameter specifies the VRF whose neighbor you want to display information about.

The *<ip-addr>* option lets you narrow the scope of the command to a specific neighbor. The display is the same as that for the command without this option except that it is limited to only the neighbor specified.

The **advertised-routes** option displays only the routes that the device has advertised to the neighbor during the current BGP4 neighbor session.

The **attribute-entries** option shows the attribute-entries associated with routes received from the neighbor.

The **flap-statistics** option shows the route flap statistics for routes received from or sent to the neighbor.

The **last-packet-with-error** option displays the last packet from the neighbor that contained an error. The packet's contents are displayed in decoded (human-readable) format.

The **received extended-community** option displays the received extended community Outbound Route Filters (ORFs) received from this neighbor.

The **received prefix-filter** option shows the Outbound Route Filters (ORFs) received from the neighbor. This option applies to cooperative route filtering.

The **routes** option lists the routes received in UPDATE messages from the neighbor. You can specify the following additional options:

- **best** – Displays the routes received from the neighbor that the device selected as the best routes to their destinations.
- **not-installed-best** – Displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
- **unreachable** – Displays the routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
- **detail** – Displays detailed information for the specified routes. You can refine your information request by also specifying one of the options above (**best**, **not-installed-best**, or **unreachable**).

The **rib-out-routes** option lists the route information base (RIB) for outbound routes. You can display all the routes or specify a network address.

The **routes-summary** option displays a summary of the following information:

- Number of routes received from the neighbor
- Number of routes accepted by this device from the neighbor

- Number of routes this device filtered out of the UPDATES received from the neighbor and did not accept
- Number of routes advertised to the neighbor
- Number of attribute entries associated with routes received from or advertised to the neighbor

This display shows the following information.

TABLE 278 BGP4 neighbor information

This field...	Displays...
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP – The neighbor is in another AS. • EBGP_Confed – The neighbor is a member of another sub-AS in the same confederation. • IBGP – The neighbor is in the same AS.
RouterID	The neighbor's ID.
Description	The description you gave the neighbor when you configured it on the device.
State	The state of the session with the neighbor. The states are from the perspective of this device of the session, not the perspective of the neighbor. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each device: • IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND – The neighbor has been administratively shut down. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. **NOTE:** If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT – BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the device receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED – BGP4 is ready to exchange UPDATE messages with the neighbor. • If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. **NOTE:** If you display information for the neighbor using the **show ip bgp neighbor** <*ip-addr*> command, the TCP receiver queue value will be greater than 0.
Time	The amount of time this session has been in its current state.

TABLE 278 BGP4 neighbor information (Continued)

This field...	Displays...
KeepAliveTime	The keep alive time, which specifies how often this device sends keep alive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the device will wait for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead.
PeerGroup	The name of the peer group the neighbor is in, if applicable.
Multihop-EBGP	Whether this option is enabled for the neighbor.
RouteReflectorClient	Whether this option is enabled for the neighbor.
SendCommunity	Whether this option is enabled for the neighbor.
NextHopSelf	Whether this option is enabled for the neighbor.
DefaultOriginate	Whether this option is enabled for the neighbor.
MaximumPrefixLimit	Lists the maximum number of prefixes the device will accept from this neighbor.
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
CooperativeFilteringCapability	Whether the neighbor is enabled for cooperative route filtering.
Distribute-list	Lists the distribute list parameters, if configured.
Filter-list	Lists the filter list parameters, if configured.
Prefix-list	Lists the prefix list parameters, if configured.
Route-map	Lists the route map parameters, if configured.
Messages Sent	The number of messages this device has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this device has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

TABLE 278 BGP4 neighbor information (Continued)

This field...	Displays...

Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following:
	• Reasons described in the BGP specifications:
	  • Message Header Error
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • OPEN Message Error
	  • Unsupported Version Number
	  • Bad Peer AS Number
	  • Bad BGP Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unsupported Capability
	  • UPDATE Message Error
	  • Malformed Attribute List
	  • Unrecognized Well-known Attribute
	  • Missing Well-known Attribute
	  • Attribute Flags Error
	  • Attribute Length Error
	  • Invalid ORIGIN Attribute
	  • Invalid NEXT_HOP Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS_PATH
	  • Hold Timer Expired
	  • Finite State Machine Error
	  • Rcv Notification

Last Connection Reset Reason (cont.)	• Reasons specific to the implementation:
	  • Reset All Peer Sessions
	  • User Reset Peer Session
	  • Port State Down
	  • Peer Removed
	  • Peer Shutdown
	  • Peer AS Number Change
	  • Peer AS Confederation Change
	  • TCP Connection KeepAlive Timeout
	  • TCP Connection Closed by Remote
	  • TCP Data Stream Error Detected

TABLE 278 BGP4 neighbor information (Continued)

This field...	Displays...

Notification Sent	If the device receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.
	• Message Header Error:
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • Unspecified
	• Open Message Error:
	  • Unsupported Version
	  • Bad Peer As
	  • Bad BGP Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unspecified
	• Update Message Error:
	  • Malformed Attribute List
	  • Unrecognized Attribute
	  • Missing Attribute
	  • Attribute Flag Error
	  • Attribute Length Error
	  • Invalid Origin Attribute
	  • Invalid NextHop Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS Path
	  • Unspecified
	• Hold Timer Expired
	• Finite State Machine Error
	• Cease
	• Unspecified

Notification Received	See above.

TABLE 278 BGP4 neighbor information (Continued)

This field...	Displays...
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the device.
Local port	The TCP port the device is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the device.
SentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the device that have not been acknowledged by the neighbor.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the device retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.

TABLE 278 BGP4 neighbor information (Continued)

This field...	Displays...
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying advertised routes for a specified VPNv4 neighbor

To display the routes the device has advertised to a specific VPNv4 neighbor, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vpnv4 neighbors 2.2.2.2 advertised-routes
     There are 231 routes advertised to neighbor 2.2.2.2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.100.100.30/32   0.0.0.0                    100        0      BE
         AS_PATH: 310
2      10.100.100.29/32   0.0.0.0                    100        0      BE
         AS_PATH: 310
3      10.100.100.28/32   0.0.0.0                    100        0      BE
         AS_PATH: 310
4      10.100.100.27/32   0.0.0.0                    100        0      BE
         AS_PATH: 310
5      10.100.100.26/32   0.0.0.0                    100        0      BE
         AS_PATH: 310
6      10.100.100.25/32   0.0.0.0                    100        0      BE
         AS_PATH: 310
7      10.100.100.24/32   0.0.0.0                    100        0      BE
         AS_PATH: 310
8      10.100.100.23/32   0.0.0.0                    100        0      BE
         AS_PATH: 310
```

Syntax: show ip bgp vpnv4 neighbor <*ip-addr*> **advertised-routes** [<*ip-addr*>/<*prefix*>]

For information about the fields in this display, refer to Table 273 on page 1621.

Displaying attribute entries for a specified VPNv4 neighbor

The neighbor attribute entries table lists the sets of BGP4 attributes stored in device memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the device typically has fewer route attribute entries than routes. To display the route-attribute entries table for a specified VPNv4 neighbor, enter the following command.

```
NetIron# show ip bgp vpnv4 neighbors 2.2.2.2 attribute-entries
        Total number of BGP Attribute Entries: 35
1       Next Hop  :0.0.0.0            Metric   :0              Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 600:1
        AS Path   :310
        Address: 0x247194b0  Hash:45 (0x0100036e) Reference Counts: 0:0:30
2       Next Hop  :0.0.0.0            Metric   :0              Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 600:1
        AS Path   :311
        Address: 0x2471a480  Hash:47 (0x01000370) Reference Counts: 0:0:30
```

Syntax: show ip bgp vpnv4 neighbors <*IPaddress*> **attribute-entries**

The <*IPaddress*> variable is the IP address of the neighbor whose attribute entries you want to display.

This display shows the following information.

TABLE 279 BGP4 route-attribute entries information

This field...	Displays...
Total number of BGP Attribute Entries	The number attribute entries in the BGP4 route table for this device.
Next Hop	The IP address of the next hop device for routes with this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • **AS Number** shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • **Router-ID** shows the device that originated this aggregator.
Router ID	

TABLE 279 BGP4 route-attribute entries information (Continued)

This field...	Displays...
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
Extended Community	The extended community attributes of the device.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	This field is for internal Dell debugging purposes only.
Hash	This field is for internal Dell debugging purposes only.
Reference Counts	This field is for internal Dell debugging purposes only.

Displaying Flap statistics for a specified VPNv4 neighbor by IP address

To display flap-statistics for routes learned from the specified VRF neighbor, enter the following command at any level of the CLI.

```
R3-2547(config)#show ip bgp vpnv4 neighbors 2.2.2.2 flap-statistics
Total number of flapping routes: 0
```

Syntax: show ip bgp vpnv4 <*vrf-name*> **neighbor** <*ip-addr*> **flap-statistics**

The <*vrf-name*> parameter specifies the VPNv4 neighbor you want to display flap-statistics for.

The <*address*> <*mask*> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

This display shows the following information.

TABLE 280 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4 route table that have changed state and thus have been marked as flapping routes.

Displaying received ORFs information for a specified VPNv4 neighbor

To view BGP4 configuration information and statistics for a specified VPNv4 neighbor, enter the following command.

```
NetIron# show ip bgp vpn neighbors 2.2.2.2 received extended-community
Extended-community ORF capability was not negotiated
 No Prefix filter ORF received from neighbor 2.2.2.2!
```

Displaying a specified neighbor VPNv4 routes

To view the route table for a specified neighbor, enter the following command.

```
NetIron#show ip bgp vpnv4 neighbors 10.10.2.3 routes
       There are 30 accepted routes from neighbor 10.10.2.3
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.100.100.1/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
2      10.100.100.2/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
3      10.100.100.3/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
4      10.100.100.4/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
5      10.100.100.5/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
6      10.100.100.6/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
7      10.100.100.7/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
8      10.100.100.8/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
9      10.100.100.9/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
```

Syntax: show ip bgp vpnv4 neighbors <*ip-addr*> **routes**

For information about the fields in this display, refer to Table 273 on page 1621.

Displaying the best routes

To display the routes received from a specific neighbor that are the “best” routes to their destinations, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vpnv4 neighbor 192.168.4.211 routes best
```

Syntax: show ip bgp vpnv4 neighbor <*ip-addr*> **routes best**

For information about the fields in this display, refer to Table 273 on page 1621.

Displaying the best routes that were nonetheless not installed in the IP route table

To display the BGP4 routes received from a specific neighbor that are the "best" routes to their destinations but are not installed in the device's IP route table, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vpnv4 neighbor 192.168.4.211 routes not-installed-best
```

Each of the displayed routes is a valid path to its destination, but the device received another path from a different source (such as OSPF, RIP, or a static route) that has a lower administrative distance. The device always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: show ip bgp vpnv4 neighbor <*ip-addr*> **routes not-installed-best**

For information about the fields in this display, refer to Table 273 on page 1621.

Displaying the routes whose destinations are unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vpnv4 neighbor 192.168.4.211 routes unreachable
```

Syntax: show ip bgp vpnv4 neighbor <*ip-addr*> **routes unreachable**

For information about the fields in this display, refer to Table 273 on page 1621.

Displaying the Adj-RIB-Out for a VRF neighbor

To display the device's current BGP4 Routing Information Base (Adj-RIB-Out) for a specific VRF neighbor and a specific destination network, enter a command such as the following at any level of the CLI.

```
NetIron#show ip bgp vpnv4 neighbor 10.10.2.3 rib-out-routes
       There are 154 RIB_out routes for neighbor 10.10.2.3
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.100.101.30/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
2      10.100.101.29/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
3      10.100.101.28/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
4      10.100.101.27/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
5      10.100.101.26/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
6      10.100.101.25/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
7      10.100.101.24/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
8      10.100.101.23/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
9      10.100.101.22/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
10     10.100.101.21/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
```

The Adj-RIB-Out contains the routes that the device either has most recently sent to the VRF neighbor or is about to send to the neighbor.

Syntax: show ip bgp vpnv4 neighbor <*ip-addr*> **rib-out-routes** [<*ip-addr*>/<*prefix*>]

For information about the fields in this display, refer to Table 273 on page 1621

Displaying routes summary for a specified VPNv4 neighbor

To view the route table for a specified VPNv4 neighbor, enter the following command.

```
NetIron#show ip bgp vpnv4 neighbor 10.10.2.3 routes-summary
1   IP Address: 10.10.2.3
Routes Accepted/Installed:30,  Filtered/Kept:0,  Filtered:0
   Routes Selected as BEST Routes:30
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0
NLRIs Received in Update Message:30,  Withdraws:0 (0),  Replacements:0
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0
Routes Advertised:154,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:154,  Withdraws:0,  Replacements:0
Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0 Outbound Routes Holder:0
```

This display shows the following information.

TABLE 281 BGP4 route summary information for a VPNv4 neighbor

This field...	Displays...
Routes Accepted or Installed	How many routes the has received from the neighbor during the current BGP4 session: • Filtered – Indicates how many of the received routes the device filtered and did not accept. • Filtered or kept – Indicates how many of the received routes the device did not accept or install because they were denied by filters.
Routes Selected as BEST Routes	The number of routes that the device selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages: • Withdraws – The number of withdrawn routes the device has received. • Replacements – The number of replacement routes the device has received.
NLRIs Discarded due to	Indicates the number of times the device discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The configured maximum prefix amount had been reached. • AS Loop – An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • Invalid Nexthop – The next hop value was not acceptable. • Duplicated Originator_ID – The originator ID was the same as the local device ID. • Cluster_ID – The cluster list contained the local cluster ID, or contained the local device ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the device has advertised to this neighbor: • To be Sent – The number of routes the device has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the device has queued up to send to this neighbor in UPDATE messages.

TABLE 281 BGP4 route summary information for a VPNv4 neighbor (Continued)

This field...	Displays...
NLRIs Sent in Update Message	The number of NLRIs for new routes the has sent to this neighbor in UPDATE messages: • Withdraws – The number of routes the device has sent to the neighbor to withdraw. • Replacements – The number of routes the device has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the device has run out of BGP4 memory for the neighbor during the current BGP4 session: • Receiving Update Messages – The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes(NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes – The number of times there was no memory for BGP4 attribute entries. • Outbound Routes (RIB-out) – The number of times there was no memory to place a "best" route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised.

Displaying summary route information

To display summary statistics for all the VPNv4 routes in the device's BGP route table, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vpnv4 routes summary
  Total number of BGP routes (NLRIs) Installed     : 184
  Distinct BGP destination networks                : 184
  Filtered bgp routes for soft reconfig            : 0
  Routes originated by this router                 : 4
  Routes selected as BEST routes                   : 184
  BEST routes not installed in IP forwarding table : 0
  Unreachable routes (no IGP route for NEXTHOP)    : 0
  IBGP routes selected as best routes              : 90
  EBGP routes selected as best routes              : 90
```

Syntax: show ip bgp vpnv4 routes summary

This display shows the following information.

TABLE 282 BGP VPNv4 summary route information

This field...	Displays...
Total number of BGP VPNv4 routes (NLRIs) Installed	The number of BGP VPNv4 routes the device has installed in the BGP route table.
Distinct BGP VPNv4 destination networks	The number of destination networks the installed routes represent. The BGP route table can have multiple routes to the same network.
Filtered BGP VPNv4 routes for soft reconfig	The number of route updates received from soft-reconfigured neighbors or peer groups that have been filtered out but retained.
Routes originated by this device	The number of VPNv4 routes in the BGP route table that this device originated.

TABLE 282 BGP VPNv4 summary route information (Continued)

This field...	Displays...
Routes selected as BEST routes	The number of VPNv4 routes in the BGP route table that this device has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	The number of BGP VPNv4 routes that are the best BGP VPNv4 routes to their destinations but were not installed in the IP route table because the device received better routes from other sources.
Unreachable routes (no IGP route for NEXTHOP)	The number of routes in the BGP route table whose destinations are unreachable because the next hop is unreachable.
IBGP routes selected as best routes	The number of "best" routes in the BGP VPNv4 route table that are IBGP routes.
EBGP routes selected as best routes	The number of "best" routes in the BGP VPNv4 route table that are EBGP routes.

Displaying the VPNv4 route table

If you want to view all the VPNv4 routes in a network, you can display the BGP VPNv4 table using the following method.

To view the BGP VPNv4 route table, enter the following command.

```
NetIron# show ip bgp vpnv4 routes
Total number of BGP Routes: 288
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
Route Distinguisher: 4:1
1      10.6.1.0/24        2.2.2.2         3          100        0      I
         AS_PATH:
2      10.8.1.0/24        2.2.2.2         2          100        0      I
         AS_PATH:
3      10.40.1.1/32       2.2.2.2         4          100        0      I
         AS_PATH:
4      10.40.1.2/32       2.2.2.2         4          100        0      I
         AS_PATH:
5      10.40.1.3/32       2.2.2.2         4          100        0      I
         AS_PATH:
```

Syntax: **show ip bgp vpnv4 routes** [<*ip-addr*>] | <*num*> | [**age** <*secs*>] | [**as-path-access-list** <*num*>] |
[**as-path-filter** <*num, num,...*>] |[**best**] | [**cidr-only**] | [**community** <*num*> | **no-export** | **no-advertise** | **internet** | **local-as**] | [**community-access-list** <*num*>] | **community-filter** <*num*> | **community-reg-expression** <*regular-expression*>
| **detail** | **local** | **neighbor** <*ip-addr*> [**next-hop** <*ip-addr*>] | [**no-best**] | [**not-installed-best**] | [**prefix-list** <*string*>] | [**regular-expression** <*regular-expression*>] | [**route-map** <*map-name*>] | [**summary**] | [**unreachable**]

The <*ip-addr*> option displays routes for a specific network.

The <*num*> option specifies the table entry with which you want the display to start. For example, if you want to list entries beginning with table entry 100, specify 100.

The **age** <*secs*> parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** <*num*> parameter filters the display using the specified AS-path ACL.

The **best** parameter displays the routes received from the neighbor that the device selected as the best routes to their destinations.

The **cidr-only** option lists only the routes whose network masks do not match their class network length.

The **community** option lets you display routes for a specific community. You can specify **local-as**, **no-export**,
no-advertise, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of up to 1– 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** <*num*> parameter filters the display using the specified community ACL.

The **community-filter** option lets you display routes that match a specific community filter.

The **community regular-expression** <*regular-expression*> option filters the display based on a specified community regular expression.

The **local** option ....

The **neighbor** <*ip-addr*> option displays the number of accepted routes from the specified BGP neighbor.

The **detail** option lets you display more details about the routes. You can refine your request by also specifying one of the other display options after the **detail** keyword.

The **next-hop** <*ip-addr*> option displays the routes for a given next-hop IP address.

The **no-best** option displays the routes for which none of the routes to a given prefix were selected as the best route.

The **not-installed-best** option displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).

The **prefix-list** <*string*> parameter filters the display using the specified IP prefix list.

The **regular-expression** <*regular-expression*> option filters the display based on a regular expression.

The **route-map** <*map-name*> parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The **summary** option displays summary information for the routes.

The **unreachable** option displays the routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.

For information about the fields in this display, refer to Table 273 on page 1621. The fields in this display also appear in the **show ip bgp vpnv4** display.

Displaying the best VPNv4 routes

To display all the VPNv4 routes in the BGP VPNv4 route table for the router that are the best routes to their destinations, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp-router)# show ip bgp vpnv4 routes best
Total number of BGP Routes: 28
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
Route Distinguisher: 4:1
1      3.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 1
3      4.60.212.0/22      192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 3356 7170 1455
5      9.2.0.0/16         192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701
```

Syntax: show ip bgp vpnv4 routes best

For information about the fields in this display, refer to Table 273 on page 1621.

Displaying best VPNv4 routes that are not in the IP route table

When the router has multiple routes to a destination, the router selects the route with the lowest administrative distance as the best route, and installs that route in the IP route table.

To display the BGP4 routes that are the “best” routes to their destinations but are not installed in the device’s IP route table, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp-router)# show ip bgp vpnv4 routes not-installed-best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Route Distinguisher: 4:1
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      BE
       AS_PATH: 65001 4355 701 80
```

Each of the displayed routes is a valid path to its destination, but the router received another path from a different source that has a lower administrative distance. The router always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: show ip bgp vpnv4 routes not-installed-best

For information about the fields in this display, refer to Table 273 on page 1621.

NOTE
To display the routes that the router has selected as the best routes and installed in the IP route table, display the IP route table using the **show ip route** command.

Displaying VPNv4 routes with unreachable destinations

To display BGP VPNv4 routes whose destinations are unreachable using any of the paths in the BGP route table, enter a command such as the following at any level of the CLI.

```
NetIron(config-bgp-router)# show ip bgp vpnv4 routes unreachable
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Route Distinguisher: 4:1
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      BE
       AS_PATH: 65001 4355 701 80
```

Syntax: show ip bgp vpnv4 routes unreachable

For information about the fields in this display, refer to Table 273 on page 1621.

Displaying information for a specific VPNv4 route

To display BGP VPNv4 route information by specifying an IP address within the network, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vpnv4 routes 10.8.1.0/24
Route Distinguisher: 4:1
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.8.1.0/24        2.2.2.2         2          100        0      I
         AS_PATH:
Route Distinguisher: 5:1
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.8.1.0/24        4.4.4.4         3          100        0      I
         AS_PATH:
```

Syntax: show ip bgp vpnv4 routes <*ip-addr*>/<*prefix*> **[longer-prefixes] |** <*ip-addr*>

For information about the fields in this display, refer to Table 273 on page 1621 and Table 283.

Displaying VPNv4 route details

Here is an example of the information displayed when you use the **detail** option. In this example, the information for one route is shown.

```
NetIron# show ip bgp vpnv4 routes detail
Total number of BGP Routes: 288
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
Route Distinguisher: 4:1
1     Prefix: 10.6.1.0/24,  Status: I,  Age: 15h36m10s
      NEXT_HOP: 2.2.2.2,  Learned from Peer: 2.2.2.2 (1)
      Out-Label: 500000
        LOCAL_PREF: 100,  MED: 3,  ORIGIN: incomplete,  Weight: 0
        AS_PATH:
        Extended Community: RT 300:1 OSPF DOMAIN ID:0.0.0.0 OSPF RT 0:1:0 OSPF RO
ID:0.0.0.0
```

For information about the fields in this display, refer to Table 273 on page 1621 and Table 283.

TABLE 283 BGP VPNv4 route information

This field...	Displays...
Prefix	The network address and prefix.
Age	The last time an update occurred.
Learned from Peer	The IP address of the neighbor that sent this route.
Local_Pref	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
MED	The route's metric. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Atomic	Whether network information in this route has been aggregated and this aggregation has resulted in information loss. **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Aggregation ID	The router that originated this aggregator.
Aggregation AS	The AS in which the network information was aggregated. This value applies only to aggregated routes and is otherwise 0.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this route has passed.
Learned From	The IP address of the neighbor from which the router learned the route.
Admin Distance	The administrative distance of the route.
Adj_RIB_out	The number of neighbors to which the route has been or will be advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4 neighbor.
Communities	The communities the route is in.
Extended Community	The routers extended community attributes.

Displaying BGP VPNv4 MPLS tag information

To display the MPLS **in-label** and **out-label** tags in the VPNv4 routes, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vpnv4 tags
    Network            Next Hop        In-Label Out-Label
Route Distinguisher: 1:1
    10.80.1.1/32       2.2.2.2         -        500003
    10.80.1.2/32       2.2.2.2         -        500003
    10.80.1.3/32       2.2.2.2         -        500003
    10.80.1.4/32       2.2.2.2         -        500003
    10.80.1.5/32       2.2.2.2         -        500003
    10.80.1.6/32       2.2.2.2         -        500003
    10.80.1.7/32       2.2.2.2         -        500003
    10.80.1.8/32       2.2.2.2         -        500003
```

The **In-Label** and **Out-Label** fields in this display are described in Table 284 below. For information about all other fields in this display, refer to Table 273 on page 1621.

TABLE 284 Route flap dampening statistics

This field...	Displays...
In-Label	Local assigned MPLS label value.
Out-Label	Learned MPLS label value

Displaying BGP or MPLS VRF information

You can display the following information about a BGP or MPLS VRF configuration on the device:

- "Displaying VRF route information"
- "Displaying VRF route information for a specified IP address"
- "Displaying attribute entries information for a specified VRF"
- "Displaying dampened paths information for a specified VRF"
- "Displaying filtered routes information for a specified VRF"
- "Displaying Flap statistics information for a specified VRF"
- "Displaying BGP neighbor information for a specified VRF"
- "Displaying advertised routes for a specified VRF neighbor"
- "Displaying neighbor attribute entries for a specified VRF"
- "Displaying flap statistics for a specified VRF neighbor by IP address"
- "Displaying received ORF information for a specified VRF neighbor"
- "Displaying received routes for a specified VRF neighbor"
- "Displaying a specified VRF neighbor routes"
- "Displaying VPNv4 routes summary for a specified VRF neighbor"
- "Displaying summary route information for a specified VRF"
- "Displaying a VRF's BGP4 route table"

Displaying VRF route information

You can display information about BGP routes that are contained within a specified VRF route table by entering a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf red
Total number of BGP Routes: 285
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop RD     Metric LocPrf Weight Path
*i  10.80.1.1/32       2.2.2.2  1:1    100    0      206 311 i
*i  10.80.1.2/32       2.2.2.2  1:1    100    0      206 311 i
*i  10.80.1.3/32       2.2.2.2  1:1    100    0      206 311 i
*i  10.80.1.4/32       2.2.2.2  1:1    100    0      206 311 i
*i  10.80.1.5/32       2.2.2.2  1:1    100    0      206 311 i
--More--, next page: Space, next line: Return key, quit: Control-c
```

This display shows the following information.

TABLE 285 BGP4 summary information

This field...	Displays...
Total number of BGP Routes:	The number of BGP routes.
Status or Status Codes	The route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE:** If the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the router received better routes from other sources (such as OSPF, RIP, or static IP routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – INTERNAL. The route was learned through BGP4. • L – LOCAL. The route originated on this router. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **NOTE:** This field appears only if you enter the **route** option.
Origin code	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.

TABLE 285 BGP4 summary information (Continued)

This field...	Displays...
RD	The Route Distinguisher. A unique ID that is prepended on any address being routed or advertised from a VRF. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" and a unique arbitrary number. For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" and a unique arbitrary number.
Network	IP address or mask of the destination network of the route.
Next Hop	The next-hop router for reaching the network from this router.
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares route on the basis of local preference, the route with the higher local preference is chosen. The preference can have a value from 0-4294967295
Weight	The value that this route associates with routes from a specific neighbor. For example, if the router receives routes to the same destination from two BGP4 neighbors, the router prefers the route from the neighbor with the larger weight.
Path	The routes AS path.

To clear the route table for a specific BGP VRF, enter the following command.

```
NetIron# clear ip bgp vrf green
```

Syntax: **clear ip bgp vrf** *<vrf-name>* **[dampening | flap-statistics | local | neighbor | routes | traffic]**

The **dampening** parameter clears route flap dampening statistics.

The **flap-statistics** parameter clears route flap statistics.

The **local** parameter clears local information.

The **neighbor** parameter clears the BGP neighbor.

The **routes** parameter clears the BGP routes.

The **traffic** parameter clears BGP traffic counters.

Displaying VRF route information for a specified IP address

To display only the routes to a specified network, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf green 10.2.2.0/24
Route Distinguisher: 2:1
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network          Next Hop      Metric LocPrf Weight Path
*i  10.2.2.0/24        4.4.4.4        1      100     0       ?
       Route is advertised to 2 peers:
        4.4.4.4(1)             2.2.2.2(1)
```

Syntax: show ip bgp vrf <*vrf-name*> <*ip-addr*>

The <*address*> <*mask*> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

The **Number of BGP Routes** matching display conditions field in this display is described in Table 286 below. For information about all other fields in this display, refer to Table 285.

TABLE 286 VRF route information

This field...	Displays...
Number of BGP Routes matching display conditions	The number of routes to the network specified as a parameter in the **show ip bgp vpnv4** <*ip-addr*> command.

Displaying attribute entries information for a specified VRF

The route-attribute entries table lists the sets of BGP attributes stored in the router's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the router typically has fewer route attribute entries than routes. To display the route-attribute entries table at any level of the CLI.

```
NetIron# show ip bgp vrf green attribute-entries
        Total number of BGP Attribute Entries: 26
1       Next Hop  :192.168.201.2      Metric   :1              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :
        Address: 0x247017ec  Hash:279 (0x03000000) Reference Counts: 1:0:0
2       Next Hop  :192.168.201.2      Metric   :2              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :
        Address: 0x247016d8  Hash:280 (0x03000000) Reference Counts: 1:0:0
3       Next Hop  :192.168.201.2      Metric   :3              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :
        Address: 0x24701900  Hash:281 (0x03000000) Reference Count
```

This display shows the following information.

TABLE 287 BGP VPNv4 attribute entries

This field...	Displays...
Total number of BGP Attribute Entries	The number of routes contained in this router's BGP4 route table.
Next Hop	The IP address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • **AS Number** shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. **Router-ID** shows the router that originated this aggregator
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	This field is used for internal Dell debugging purposes only.
Hash	This field is used for internal Dell debugging purposes only.
Reference Counts	This field is used for internal Dell debugging purposes only.

Displaying dampened paths information for a specified VRF

To view BGP VPNv4 paths suppressed due to dampening for a specified VRF, enter the following command.

```
NetIron# show ip bgp vrf green dampened-paths
```

Displaying filtered routes information for a specified VRF

To view BGP VPNv4 filtered paths information for a specified VRF, enter the following command.

```
NetIron# show ip bgp vrf green filtered-routes
```

Displaying Flap statistics information for a specified VRF

To display flap statistics for a specified VRF, enter the following command at any level of the CLI.

```
NetIron# show ip bgp vrf green flap-statistics
```

Syntax: **show ip bgp flap-statistics [regular-expression** <*regular-expression*> **|** <*address*> <*mask*> **[longer-prefixes] | neighbor** <*ip-addr*> **| filter-list** <*num*>**...]**

The <*address*> <*mask*> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

The **as-path-filter** <*num*> parameter specifies one or more filters. Only the routes that have been dampened and that match the specified filter or filters are displayed.

The **neighbor** <*ip-addr*> parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbor** <*ip-addr*> **flap-statistics**.

The **regular-expression** <*regular-expression*> parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters.

This display shows the following information.

TABLE 288 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4 route table that have changed state and thus have been marked as flapping routes.

You also can display all the dampened routes by entering the following command: **show ip bgp dampened-paths**.

Displaying BGP neighbor information for a specified VRF

To view BGP4 configuration information and statistics for a specified VRF's neighbors, enter the following command.

```
NetIron# show ip bgp vrf black neighbor
    Total number of BGP Neighbors: 3
1   IP Address: 10.10.2.3, AS: 206 (EBGP), RouterID: 10.10.2.3, VRF: black
    State: ESTABLISHED, Time: 14h31m51s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 4 seconds, HoldTimer Expire in 135 seconds
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       4       871       0            0
       Received: 1       1       873       0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  unicast  capability
      Peer configured for IPV4 unicast  Routes
    TCP Connection state: ESTABLISHED
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:   17543, Received: 16814
       Local host:  10.10.2.1, Local  Port: 8135
       Remote host: 10.10.2.3, Remote Port: 179
       ISentSeq:    3301937  SendNext:    3319481  TotUnAck:          0
       TotSent:       17544  ReTrans:           0  UnAckSeq:    3319481
       IRcvSeq:   466270178  RcvNext:   466286993  SendWnd:        6432
       TotalRcv:      16815  DupliRcv:        285  RcvWnd:        65000
       SendQue:           0  RcvQue:            0  CngstWnd:       1456
```

This example shows how to display information for a specific VRF's neighbor. None of the other display options are used; thus, all of the information is displayed for the neighbor. The number in the far left column indicates the neighbor for which information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the device's Transmission Control Block (TCB) for the TCP session between the router and its neighbor. These fields are described in detail in section *3.2 of RFC 793*, "*Transmission Control Protocol Functional Specification*".

Syntax: **show ip bgp vrf** <*vrf-name*> **neighbors** [<*ip-addr*> **[advertised-routes [detail** [<*ip-addr*>[/<*mask-bits*>]]]] **|**
[attribute-entries [detail]] | [flap-statistics] | [last-packet-with-error] | [received prefix-filter] |
[received-routes] | [routes [best] | [detail [best] | [not-installed-best] | [unreachable]] |
[rib-out-routes [<*ip-addr*>/<*mask-bits*> **|** <*ip-addr*> <*net-mask*> **| detail]] |**
[routes-summary]]

The <*vrf-name*> parameter specifies the VRF whose neighbor you want to display information about.

The <*ip-addr*> option lets you narrow the scope of the command to a specific neighbor. The display is the same as that for the command without this option except that it is limited to only the neighbor specified.

The **advertised-routes** option displays only the routes that the router has advertised to the neighbor during the current BGP4 neighbor session.

The **attribute-entries** option shows the attribute-entries associated with routes received from the neighbor.

The **flap-statistics** option shows the route flap statistics for routes received from or sent to the neighbor.

The **last-packet-with-error** option displays the last packet from the neighbor that contained an error. The packet's contents are displayed in decoded (human-readable) format.

The **received extended-community** option

The **received prefix-filter** option shows the Outbound Route Filters (ORFs) received from the neighbor. This option applies to cooperative route filtering.

The **received-routes** option lists all the route information received in route updates from the neighbor since the soft reconfiguration feature was enabled.

The **routes** option lists the routes received in UPDATE messages from the neighbor. You can specify the following additional options:

- **best** – Displays the routes received from the neighbor that the router selected as the best routes to their destinations.
- **not-installed-best** – Displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the router received better routes from other sources (such as OSPF, RIP, or static IP routes).
- **unreachable** – Displays the routes that are unreachable because the router does not have a valid RIP, OSPF, or static route to the next hop.
- **detail** – Displays detailed information for the specified routes. You can refine your information request by also specifying one of the options above (best, not-installed-best, or unreachable).

The **rib-out-routes** option lists the route information base (RIB) for outbound routes. You can display all the routes or specify a network address.

The **routes-summary** option displays a summary of the following information:

- Number of routes received from the neighbor
- Number of routes accepted by this router from the neighbor
- Number of routes this router filtered out of the UPDATES received from the neighbor and did not accept
- Number of routes advertised to the neighbor
- Number of attribute entries associated with routes received from or advertised to the neighbor.

This display shows the following information.

TABLE 289 BGP4 neighbor information

This field...	Displays...
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.

TABLE 289 BGP4 neighbor information (Continued)

This field...	Displays...
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP – The neighbor is in another AS. • EBGP_Confed – The neighbor is a member of another sub-AS in the same confederation. • IBGP – The neighbor is in the same AS.
RouterID	The neighbor's router ID.
Description	The description you gave the neighbor when you configured it on the router.

State	The state of the router's session with the neighbor. The states are from this router's perspective of the session, not the neighbor's perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each router:
	• IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process.
	  • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• ADMND – The neighbor has been administratively shut down.
	  • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed.
	• ACTIVE – BGP4 is waiting for a TCP connection from the neighbor.
	NOTE: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.
	• OPEN SENT – BGP4 is waiting for an Open message from the neighbor.
	• OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the router receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle.
	• ESTABLISHED – BGP4 is ready to exchange UPDATE messages with the neighbor.
	  • If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed.
	NOTE: If you display information for the neighbor using the **show ip bgp neighbor** *<ip-addr>* command, the TCP receiver queue value will be greater than 0.

Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this router sends keep alive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the router will wait for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead.
PeerGroup	The name of the peer group the neighbor is in, if applicable.
Multihop-EBGP	Whether this option is enabled for the neighbor.
RouteReflectorClient	Whether this option is enabled for the neighbor.
SendCommunity	Whether this option is enabled for the neighbor.
NextHopSelf	Whether this option is enabled for the neighbor.
DefaultOriginate	Whether this option is enabled for the neighbor.

TABLE 289 BGP4 neighbor information (Continued)

This field...	Displays...
MaximumPrefixLimit	Lists the maximum number of prefixes the router will accept from this neighbor.
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this router has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
CooperativeFilteringCapability	Whether the neighbor is enabled for cooperative route filtering.
Distribute-list	Lists the distribute list parameters, if configured.
Filter-list	Lists the filter list parameters, if configured.
Prefix-list	Lists the prefix list parameters, if configured.
Route-map	Lists the route map parameters, if configured.
Messages Sent	The number of messages this router has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this router has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

TABLE 289 BGP4 neighbor information (Continued)

This field...	Displays...

Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following:
	• Reasons described in the BGP specifications:
	  • Message Header Error
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • OPEN Message Error
	  • Unsupported Version Number
	  • Bad Peer AS Number
	  • Bad BGP Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unsupported Capability
	  • UPDATE Message Error
	  • Malformed Attribute List
	  • Unrecognized Well-known Attribute
	  • Missing Well-known Attribute
	  • Attribute Flags Error
	  • Attribute Length Error
	  • Invalid ORIGIN Attribute
	  • Invalid NEXT_HOP Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS_PATH
	  • Hold Timer Expired
	  • Finite State Machine Error
	  • Rcv Notification

Last Connection Reset Reason (cont.)	• Reasons specific to the implementation:
	  • Reset All Peer Sessions
	  • User Reset Peer Session
	  • Port State Down
	  • Peer Removed
	  • Peer Shutdown
	  • Peer AS Number Change
	  • Peer AS Confederation Change
	  • TCP Connection KeepAlive Timeout
	  • TCP Connection Closed by Remote
	  • TCP Data Stream Error Detected

TABLE 289 BGP4 neighbor information (Continued)

This field...	Displays...

Notification Sent	If the router receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.
	• Message Header Error:
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • Unspecified
	• Open Message Error:
	  • Unsupported Version
	  • Bad Peer As
	  • Bad BGP Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unspecified
	• Update Message Error:
	  • Malformed Attribute List
	  • Unrecognized Attribute
	  • Missing Attribute
	  • Attribute Flag Error
	  • Attribute Length Error
	  • Invalid Origin Attribute
	  • Invalid NextHop Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS Path
	  • Unspecified
	• Hold Timer Expired
	• Finite State Machine Error
	• Cease
	• Unspecified

Notification Received	See above.

TABLE 289 BGP4 neighbor information (Continued)

This field...	Displays...
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the router.
Local port	The TCP port the router is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the router.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the router that have not been acknowledged by the neighbor.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the router retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.

TABLE 289 BGP4 neighbor information (Continued)

This field...	Displays...
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying advertised routes for a specified VRF neighbor

To display the routes the router has advertised to a specific VRF's neighbor, enter a command such as the following at any level of the CLI.

```
R3-2547#show ip bgp vrf black neighbor 10.10.2.3 advertised-routes
       There are 154 routes advertised to neighbor 10.10.2.3
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.100.101.30/32   10.10.3.3                             0      BE
         AS_PATH: 311
2      10.100.101.29/32   10.10.3.3                             0      BE
         AS_PATH: 311
3      10.100.101.28/32   10.10.3.3                             0      BE
         AS_PATH: 311
4      10.100.101.27/32   10.10.3.3                             0      BE
         AS_PATH: 311
5      10.100.101.26/32   10.10.3.3                             0      BE
         AS_PATH: 311
6      10.100.101.25/32   10.10.3.3                             0      BE
         AS_PATH: 311
7      10.100.101.24/32   10.10.3.3                             0      BE
         AS_PATH: 311
8      10.100.101.23/32   10.10.3.3                             0      BE
         AS_PATH: 311
9      10.100.101.22/32   10.10.3.3                             0      BE
         AS_PATH: 311
10     10.100.101.21/32   10.10.3.3                             0      BE
         AS_PATH: 311
```

Syntax: show ip bgp vrf <*vrf-name*> **neighbor** <*ip-addr*> **advertised-routes** [<*ip-addr*>/<*prefix*>]

For information about the fields in this display, refer to Table 273 on page 1621.

Displaying neighbor attribute entries for a specified VRF

The neighbor attribute entries table lists the sets of BGP4 attributes stored in the router's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the router typically has fewer route attribute entries than routes. To display the route-attribute entries table for a specified VRF, enter the following command.

```
NetIron#show ip bgp vrf black neighbor 10.10.2.3 attribute-entries
        Total number of BGP Attribute Entries: 2
1       Next Hop  :10.10.2.3          Metric   :0              Origin:IGP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :310
        Address: 0x2470139c  Hash:223 (0x0100036e) Reference Counts: 30:0:60
2       Next Hop  :2.2.2.2            Metric   :2              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 600:1 OSPF DOMAIN ID:0.0.0.0 OSPF RT 0:5:1 OSPF RO
ID:0.0.0.0
        AS Path   :
        Address: 0x24702310  Hash:992 (0x03000000) Reference Counts: 0:0:90
```

Syntax: show ip bgp <*vrf-name*> **attribute-entries**

This display shows the following information.

TABLE 290 BGP4 route-attribute entries information

This field...	Displays...
Total number of BGP Attribute Entries	The number of routes contained in this router's BGP4 route table.
Next Hop	The IP address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • **AS Number** shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • **Router-ID** shows the router that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.

TABLE 290 BGP4 route-attribute entries information (Continued)

This field...	Displays...
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
Extended Community	The VRF's extended community attibutes.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.

Displaying flap statistics for a specified VRF neighbor by IP address

To display flap-statistics for routes learned from the specified VRF neighbor, enter the following command at any level of the CLI.

```
R3-2547#show ip bgp vrf black neighbor 10.10.2.3 flap-statistics
Total number of flapping routes: 0
```

Syntax: show ip bgp vrf <*vrf-name*> **neighbor** <*ip-addr*> **flap-statistics**

The <*vrf-name*> parameter specifies the VRF whose neighbor you want to display flap-statistics for.

The <*address*> <*mask*> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

This display shows the following information.

TABLE 291 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4 route table that have changed state and thus have been marked as flapping routes.

Displaying received ORF information for a specified VRF neighbor

To view BGP4 VPNv4 configuration information and statistics for a specified VRF's neighbor, enter the following command.

```
NetIron #show ip bgp vrf black neighbor 10.10.2.3 received extended-community
Extended-community ORF capability was not negotiated

NetIron#show ip bgp vrf black neighbor 10.10.2.3 received prefix-filter
 No Prefix filter ORF received from neighbor 10.10.2.3!
```

Displaying received routes for a specified VRF neighbor

To view the BGP4 VPNv4 configuration and statistics for specified VRF's neighbor, enter the following command.

```
NetIron#show ip bgp vrf black neighbor 10.10.2.3 received-routes
Inbound soft reconfiguration not enabled for neighbor 10.10.2.3
```

Displaying a specified VRF neighbor routes

To view the route table for a specified VRF's neighbor, enter the following command.

```
NetIron#show ip bgp vrf black neighbor 10.10.2.3 routes
       There are 30 accepted routes from neighbor 10.10.2.3
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.100.100.1/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
2      10.100.100.2/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
3      10.100.100.3/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
4      10.100.100.4/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
5      10.100.100.5/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
6      10.100.100.6/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
7      10.100.100.7/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
8      10.100.100.8/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
9      10.100.100.9/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
```

Syntax: show ip bgp vrf <*vrf-name*> neighbor <*ip-addr*> routes

For information about the fields in this display, refer to Table 285 on page 1647.

Displaying the best routes

To display the routes received from a specific neighbor that are the "best" routes to their destinations, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf black neighbor 192.168.4.211 routes best
```

Syntax: show ip bgp vrf <*vrf-name*> **neighbor** <*ip-addr*> **routes best**

For information about the fields in this display, refer to Table 285 on page 1647.

Displaying the best routes that were nonetheless not installed in the IP route table

To display the BGP4 routes received from a specific neighbor that are the "best" routes to their destinations but are not installed in the router's IP route table, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf black neighbor 192.168.4.211 routes not-installed-best
```

Each of the displayed routes is a valid path to its destination, but the router received another path from a different source (such as OSPF, RIP, or a static route) that has a lower administrative distance. The router always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: show ip bgp vrf <*vrf-name*> **neighbor** <*ip-addr*> **routes not-installed-best**

For information about the fields in this display, refer to Table 285 on page 1647.

Displaying the routes whose destinations are unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf black neighbor 192.168.4.211 routes unreachable
```

Syntax: show ip bgp vrf <*vrf-name*> **neighbor** <*ip-addr*> **routes unreachable**

For information about the fields in this display, refer to Table 285 on page 1647.

Displaying the Adj-RIB-Out for a VRF neighbor

To display the router's current BGP4 Routing Information Base (Adj-RIB-Out) for a specific VRF neighbor and a specific destination network, enter a command such as the following at any level of the CLI.

```
NetIron#show ip bgp vrf black neighbor 10.10.2.3 rib-out-routes
       There are 154 RIB_out routes for neighbor 10.10.2.3
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.100.101.30/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
2      10.100.101.29/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
3      10.100.101.28/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
4      10.100.101.27/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
5      10.100.101.26/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
6      10.100.101.25/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
7      10.100.101.24/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
8      10.100.101.23/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
9      10.100.101.22/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
10     10.100.101.21/32   10.10.3.3                  100        0      BE
         AS_PATH: 311
```

The Adj-RIB-Out contains the routes that the router either has most recently sent to the VRF neighbor or is about to send to the neighbor.

Syntax: show ip bgp vrf <*vrf-name*> **neighbor** <*ip-addr*> **rib-out-routes** [<*ip-addr*>/<*prefix*>]

For information about the fields in this display, refer to Table 285 on page 1647.

Displaying VPNv4 routes summary for a specified VRF neighbor

To view the route table for a specified VRF's neighbor, enter the following command.

```
NetIron#show ip bgp vrf black neighbor 10.10.2.3 routes-summary
1   IP Address: 10.10.2.3
Routes Accepted/Installed:30,  Filtered/Kept:0,  Filtered:0
   Routes Selected as BEST Routes:30
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0
NLRIs Received in Update Message:30,  Withdraws:0 (0),  Replacements:0
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0
Routes Advertised:154,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:154,  Withdraws:0,  Replacements:0
Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0 Outbound Routes Holder:0
```

This display shows the following information.

TABLE 292 BGP4 route summary information for a VRF neighbor

This field...	Displays...
Routes Received	How many routes the router has received from the neighbor during the current BGP4 session. • Accepted or Installed – Indicates how many of the received routes the router accepted and installed in the BGP4 route table. • Filtered – Indicates how many of the received routes the device did not accept or install because they were denied by filters on the router.
Routes Selected as BEST Routes	The number of routes that the router selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the router received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the router does not have a valid RIP, OSPF, or static route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages. • Withdraws – The number of withdrawn routes the router has received. • Replacements – The number of replacement routes the router has received.
NLRIs Discarded due to	Indicates the number of times the router discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The router's configured maximum prefix amount had been reached. • AS Loop – An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • Invalid Nexthop – The next hop value was not acceptable. • Duplicated Originator_ID – The originator ID was the same as the local router ID. • Cluster_ID – The cluster list contained the local cluster ID, or contained the local router ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the router has advertised to this neighbor: • To be Sent – The number of routes the router has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the router has queued up to send to this neighbor in UPDATE messages.

TABLE 292 BGP4 route summary information for a VRF neighbor (Continued)

This field...	Displays...
NLRIs Sent in Update Message	The number of NLRIs for new routes the router has sent to this neighbor in UPDATE messages: • Withdraws – The number of routes the router has sent to the neighbor to withdraw. • Replacements – The number of routes the router has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the router has run out of BGP4 memory for the neighbor during the current BGP4 session: • Receiving Update Messages – The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes(NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes – The number of times there was no memory for BGP4 attribute entries. • Outbound Routes (RIB-out) – The number of times there was no memory to place a "best" route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised.

Displaying summary route information for a specified VRF

To display summary statistics for all the VPNv4 routes in the device's BGP route table for a specified VRF, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf black routes summary
  Total number of BGP routes (NLRIs) Installed     : 184
  Distinct BGP destination networks                : 184
  Filtered bgp routes for soft reconfig            : 0
  Routes originated by this router                 : 4
  Routes selected as BEST routes                   : 184
  BEST routes not installed in IP forwarding table : 0
  Unreachable routes (no IGP route for NEXTHOP)    : 0
  IBGP routes selected as best routes              : 90
  EBGP routes selected as best routes              : 90
```

Syntax: show ip bgp vrf <*vrf-name*> **routes summary**

This display shows the following information.

TABLE 293 BGP VPNv4 summary route information

This field...	Displays...
Total number of BGP VPNv4 routes (NLRIs) Installed	The number of BGP VPNv4 routes the router has installed in the BGP route table.
Distinct BGP VPNv4 destination networks	The number of destination networks the installed routes represent. The BGP route table can have multiple routes to the same network.
Filtered BGP VPNv4 routes for soft reconfig	The number of route updates received from soft-reconfigured neighbors or peer groups that have been filtered out but retained.

TABLE 293 BGP VPNv4 summary route information (Continued)

This field...	Displays...
Routes originated by this router	The number of VPNv4 routes in the BGP route table that this router originated.
Routes selected as BEST routes	The number of VPNv4 routes in the BGP route table that this router has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	The number of BGP VPNv4 routes that are the best BGP VPNv4 routes to their destinations but were not installed in the IP route table because the router received better routes from other sources.
Unreachable routes (no IGP route for NEXTHOP)	The number of routes in the BGP route table whose destinations are unreachable because the next hop is unreachable.
IBGP routes selected as best routes	The number of "best" routes in the BGP VPNv4 route table that are IBGP routes.
EBGP routes selected as best routes	The number of "best" routes in the BGP VPNv4 route table that are EBGP routes.

Displaying a VRF's BGP4 route table

If you want to view all the routes BGP routes in a VRF, you can display the VRF's BGP route table using the following method.

To view a VRF's BGP4 route table, enter the following command.

```
NetIron# show ip bgp vrf black routes
Total number of BGP Routes: 184
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      7.7.7.7/32         0.0.0.0         0          100        32768  BL
         AS_PATH:
2      10.10.2.0/24       0.0.0.0         0          100        32768  BL
         AS_PATH:
3      10.10.3.0/24       0.0.0.0         0          100        32768  BL
         AS_PATH:
4      10.10.4.0/24       0.0.0.0         0          100        32768  BL
         AS_PATH:
5      10.100.100.1/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
6      10.100.100.2/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
7      10.100.100.3/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
8      10.100.100.4/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
9      10.100.100.5/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
10     10.100.100.6/32    10.10.2.3                  100        0      BE
```

Syntax: **show ip bgp vrf** <*vrf-name*> **routes** [<*ip-addr*>] | <*num*> | [**age** <*secs*>] | [**as-path-access-list** <*num*>] | [**as-path-filter** <*num, num,...*>] |[**best**] | [**cidr-only**] | [**community** <*num*> | **no-export** | **no-advertise** | **internet** | **local-as**] | [**community-access-list** <*num*>] | **community-filter** <*num*> | **community-reg-expression** <*regular-expression*>

| detail | local | neighbor <*ip-addr*> **[next-hop** <*ip-addr*>**] | [no-best] | [not-installed-best] | [prefix-list** <*string*>**] | [regular-expression** <*regular-expression*>**] | [route-map** <*map-name*>**] | [summary] | [unreachable]**

The <*vrf-name*> parameter specifies the VRF whose neighbor you want to display information about.

The <*ip-addr*> option displays routes for a specific network.

The <*num*> option specifies the table entry with which you want the display to start. For example, if you want to list entries beginning with table entry 100, specify 100.

The **age** <*secs*> parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** <*num*> parameter filters the display using the specified AS-path ACL.

The **best** parameter displays the routes received from the neighbor that the router selected as the best routes to their destinations.

The **cidr-only** option lists only the routes whose network masks do not match their class network length.

The **community** option lets you display routes for a specific community. You can specify **local-as**, **no-export**, **no-advertise**, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of up to 1– 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** <*num*> parameter filters the display using the specified community ACL.

The **community-filter** option lets you display routes that match a specific community filter.

The c**ommunity regular-expression** <*regular-expression*> option filters the display based on a specified community regular expression.

The **local** option ....

The **neighbor** <*ip-addr*> option displays the number of accepted routes from the specified BGP neighbor.

The **detail** option lets you display more details about the routes. You can refine your request by also specifying one of the other display options after the **detail** keyword.

The **next-hop** <*ip-addr*> option displays the routes for a given next-hop IP address.

The **no-best** option displays the routes for which none of the routes to a given prefix were selected as the best route.

The **not-installed-best** option displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the router received better routes from other sources (such as OSPF, RIP, or static IP routes).

The **prefix-list** <*string*> parameter filters the display using the specified IP prefix list.

The **regular-expression** <*regular-expression*> option filters the display based on a regular expression.

The **route-map** <*map-name*> parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The **summary** option displays summary information for the routes.

The **unreachable** option displays the routes that are unreachable because the router does not have a valid RIP, OSPF, or static route to the next hop.

For information about the fields in this display, refer to Table 273 on page 1621. The fields in this display also appear in the **show ip bgp vpnv4** display.

Displaying the best BGP4 routes

To display all the BGP4 routes in the device's BGP4 route table that are the best routes to their destinations, enter a command such as the following at any level of the CLI.

```
NetIron#show ip bgp vrf black routes best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      7.7.7.7/32         0.0.0.0         0          100        32768  BL
         AS_PATH:
2      10.10.2.0/24       0.0.0.0         0          100        32768  BL
         AS_PATH:
3      10.10.3.0/24       0.0.0.0         0          100        32768  BL
         AS_PATH:
4      10.10.4.0/24       0.0.0.0         0          100        32768  BL
         AS_PATH:
5      10.100.100.1/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
6      10.100.100.2/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
7      10.100.100.3/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
8      10.100.100.4/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
9      10.100.100.5/32    10.10.2.3                  100        0      BE
         AS_PATH: 310
```

Syntax: show ip bgp vrf <*vrf-name*> **routes best**

For information about the fields in this display, refer to Table 285 on page 1647.

Displaying best BGP4 routes that are not in the IP route table

When the router has multiple routes to a destination, the router selects the route with the lowest administrative distance as the best route, and installs that route in the IP route table.

To display the BGP4 routes for a specified VRF that are the "best" routes to their destinations but are not installed in the device's IP route table, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf black routes not-installed-best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Route Distinguisher: 4:1
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      BE
       AS_PATH: 65001 4355 701 80
```

Each of the displayed routes is a valid path to its destination, but the router received another path from a different source that has a lower administrative distance. The router always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: show ip bgp vrf <*vrf-name*> **routes not-installed-best**

For information about the fields in this display, refer to Table 285 on page 1647.

NOTE
To display the routes that the router has selected as the best routes and installed in the IP route table, display the IP route table using the **show ip route** command.

Displaying BGP4 routes whose destinations are unreachable

To display BGP routes for a specified VRF whose destinations are unreachable using any of the paths in the BGP route table, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf black routes unreachable
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Route Distinguisher: 4:1
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      BE
       AS_PATH: 65001 4355 701 80
```

Syntax: show ip bgp vrf black routes unreachable

For information about the fields in this display, refer to Table 285 on page 1647.

Displaying information for a specific route

To display BGP VPNv4 route information for a specified VRF by specifying an IP address within the network, enter a command such as the following at any level of the CLI.

```
NetIron# show ip bgp vrf black routes 10.8.1.0/24
Route Distinguisher: 4:1
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.8.1.0/24        2.2.2.2         2          100        0      I
         AS_PATH:
Route Distinguisher: 5:1
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.8.1.0/24        4.4.4.4         3          100        0      I
         AS_PATH:
```

Syntax: show ip bgp vrf black routes <*ip-addr*>/<*prefix*> **[longer-prefixes] |** <*ip-addr*>

For information about the fields in this display, refer to Table 285 on page 1647 and Table 283 on page 1645.

Displaying route details

Here is an example of the information displayed when you use the **detail** option. In this example, the information for one route is shown.

```
NetIron# show ip bgp vrf black routes detail
Total number of BGP Routes: 288
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
Route Distinguisher: 4:1
1     Prefix: 10.6.1.0/24,  Status: I,  Age: 15h36m10s
      NEXT_HOP: 2.2.2.2,  Learned from Peer: 2.2.2.2 (1)
      Out-Label: 500000
        LOCAL_PREF: 100,  MED: 3,  ORIGIN: incomplete,  Weight: 0
        AS_PATH:
        Extended Community: RT 300:1 OSPF DOMAIN ID:0.0.0.0 OSPF RT 0:1:0 OSPF RO
ID:0.0.0.0
```

For information about the fields in this display, refer to Table 285 on page 1647 and Table 294.

TABLE 294 BGP VPNv4 route information

This field...	Displays...
Prefix	The network address and prefix.
Age	The last time an update occurred.
Learned from Peer	The IP address of the neighbor that sent this route.
Local_Pref	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
MED	The route's metric. If the route has no metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Atomic	Whether network information in this route has been aggregated and this aggregation has resulted in information loss. **NOTE:** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Aggregation ID	The router that originated this aggregator.
Aggregation AS	The AS in which the network information was aggregated. This value applies only to aggregated routes and is otherwise 0.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this route has passed.
Learned From	The IP address of the neighbor from which the router learned the route.
Admin Distance	The administrative distance of the route.

TABLE 294 BGP VPNv4 route information (Continued)

This field...	Displays...
Adj_RIB_out	The number of neighbors to which the route has been or will be advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4 neighbor.
Communities	The communities the route is in.
Extended Community	The extended communities the route is in.

Displaying additional BGP or MPLS VPN information

You can display the following additional information about a BGP or MPLS configuration on the device:

- "Displaying IP network information for a VRF"
- "Displaying the IP route table for a specified VRF"
- "Displaying ARP VRF information"
- "Displaying OSPF information for a VRF"
- "Displaying OSPF area information for a VRF"
- "Displaying OSPF ABR and ASBR information for a VRF"
- "Displaying general OSPF configuration information for a VRF"
- "Displaying OSPF external link state information for a VRF"
- "Displaying OSPF interface information"
- "Displaying OSPF neighbor information for a VRF"
- "Displaying the routes that have been redistributed into OSPF"
- "Displaying OSPF route information for a VRF"
- "Displaying OSPF sham links"
- "Displaying OSPF trap status for a VRF"
- "Displaying OSPF virtual links for a VRF"
- "Displaying OSPF virtual neighbor information for a VRF"
- "Displaying IP extcommunity list information"
- "Displaying the IP static route table for a VRF"
- "Displaying the static ARP table for a VRF"
- "Displaying TCP connections for a VRF"
- "Displaying MPLS statistics for a VRF"

Displaying VRF information

To display IP Information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron# show ip vrf green
VRF green, default RD 4:1, Table ID 2
Label: 500000,  Label-Switched Mode: OFF
Max Routes: 64000
  Interfaces:
    e6/3 lo4 lo5 lo6
  Export VPN route-target communities:
    RT:300:1
  Import VPN route-target communities:
    RT:300:1
  No import route-map
  Export route-map: export1
```

Syntax: show ip vrf <*vrf-name*>

The <*vrf-name*> parameter specifies the VRF that you want to display IP information for.

TABLE 295 Output from the show IP VRF command

This field...	Displays...
VRF Name	The name of the VRF.
Default RD	The default route distinguisher for the VRF.
Table ID	The table ID for the VRF.
Label	Display the unique VRF label that has been assigned to the specified VRF.
Label Switched Mode	Displays if Label Switched Mode is ON or OFF.
Max routes	The maximum number of routes that can be configured on this VRF.
Interfaces	The interface from this router that are configured within this VRF.
Export VPN route-target communities:	The export route-targets that are configured for this VRF.
Import VPN route-target communities	The import route-targets that are configured for this VRF.
Import route-map	The name of the import route-map if any that is configured for this VRF.
Export route-map	The name of the export route-map if a route-map has been configured for this VRF.

Displaying IP network information for a VRF

To display IP network information for a specified VRF, use the following command at any level of the CLI.

```
NetIron# show ip network vrf green
Total IP and IPVPN Cache Entry Usage on LPs:
 Module        Host    Network       Free      Total
      2          26          0     204774     204800
      5          28        240     204532     204800
```

Syntax: show ip network vrf <*vrf-name*>

This display shows the following information.

TABLE 296 BGP VPNv4 summary route information

This field...	Displays...
Module	The slot number of the module.
Host	The number of host cache entries.
Network	The number of network cache entries
Free	The number of cache entries that are unused.
Total	The total number of cache entries used and unused.

Displaying the IP route table for a specified VRF

To display the IP routes for a specified VRF, enter the following command at any CLI level.

```
NetIron#show ip route vrf green
Total number of IP routes: 99
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
        Destination        Gateway         Port        Cost       Type
1       10.5.1.0/24        192.168.201.2   eth 6/3     110/2      O
2       10.6.1.0/24        4.4.4.4         lsp toR4    200/0      B
3       10.8.1.0/24        2.2.2.2         lsp toR2    200/0      B
4       10.30.1.1/32       192.168.201.2   eth 6/3     110/3      O1
5       10.30.1.2/32       192.168.201.2   eth 6/3     110/3      O1
6       10.30.1.3/32       192.168.201.2   eth 6/3     110/3      O1
7       10.30.1.4/32       192.168.201.2   eth 6/3     110/3      O1
8       10.30.1.5/32       192.168.201.2   eth 6/3     110/3      O1
9       10.30.1.6/32       192.168.201.2   eth 6/3     110/3      O1
10      10.30.1.7/32       192.168.201.2   eth 6/3     110/3      O1
11      10.30.1.8/32       192.168.201.2   eth 6/3     110/3      O1
```

Syntax: show ip route vrf <*vrf-name*>

The <*vrf-name*> parameter specifies the VRF that you want to display IP routes for.

The following table lists the information displayed by the **show ip route vrf** command.

TABLE 297 CLI display of IP route table

This field...	Displays...
Total number of IP routes	The total number of IP routes that are in the specified VRP routing table.
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router.
Port	The port through which this router sends packets to reach the route's destination.

TABLE 297 CLI display of IP route table (Continued)

This field...	Displays...
Cost	The route's cost.

Type	The route type, which can be one of the following:
	• B – The route was learned from BGP.
	• D – The destination is directly connected to this router.
	• R – The route was learned from RIP.
	• S – The route is a static route.
	• * – The route is a candidate default route.
	• O – The route is an OSPF route. Unless you use the **ospf** option to display the route table, "O" is used for all OSPF routes. If you do use the **ospf** option, the following type codes are used:
	  • O – OSPF intra area route (within the same area).
	  • IA – The route is an OSPF inter area route (a route that passes from one area into another).
	  • E1 – The route is an OSPF external type 1 route.
	  • E2 – The route is an OSPF external type 2 route.

Displaying ARP VRF information

To display the ARP information for a specified VRF, enter the following command.

```
NetIron#show arp vrf green
Total number of ARP entries: 9
Entries in VRF green:
      IP Address          MAC Address         Type        Age      Port
1     192.168.201.2       00e0.52cf.e840      Dynamic     0         6/3
```

Syntax: show arp vrf <*vrf-name*> **[number] [ip-address] [ethernet** <*slot/port*>**] [mac-address** <*mac-addr*>**]**

The <*vrf-name*> parameter specifies the VRF that you want to display arp entries for.

To clear the ARP table.

```
NetIron# clear arp vrf green
```

Syntax: clear arp vrf <*vrf-name*>

Displaying OSPF information for a VRF

To display the OSPF Information for a specified VRF, enter the following command at any CLI level.

```
NetIron#show ip ospf vrf green
OSPF Version Number           Version 2
Router Id                     192.168.201.1
Domain Id                     2.2.2.2
Domain Tag                    2.2.2.2
ASBR Status                   Yes
ABR Status                    Yes        (1)
Redistribute Ext Routes from  BGP
External LSA Counter          96
Originate New LSA Counter     1738
Rx New LSA Counter            173
External LSA Limit            14447047
Database Overflow Interval    0
Database Overflow State :     NOT OVERFLOWED
RFC 1583 Compatibility :        Enabled
```

Syntax: show ip ospf vrf <*vrf-name*> **[area** [<*area-id*> **|** <*area-ipaddress*>**]] [border-routers** <*router-id*>**] [config] [database [database-summary | external-link-state [advertise** <*number*>**] | extensive | link-state-id** <*id-number*> **|router-id** <*advertising-router-id*> **| sequence-number** <*HEX*>**] [link-state [advertise** <*number*>**] | sabre <**

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF information for.

Displaying OSPF area information for a VRF

To display OSPF Area Information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron#show ip ospf vrf green area
Indx Area            Type   Cost       SPFR       ABR   ASBR  LSA   Chksum(Hex)
1    0               normal 0          6          0     0     6     00039ba2
2    1               normal 0          6          0     2     6     0003af4b
```

Syntax: show ip ospf vrf <*vrf-name*> **area** [<*area-id*>] **|** [<*ip-address*>]

The <*vrf-name*> parameter specifies the VRF that you want to the OSPF area information for.

The <*area-id*> parameter shows information for the specified area.

The <*ip-address*> parameter displays the entry that corresponds to the IP address you enter.

Displaying OSPF ABR and ASBR information for a VRF

To display OSPF ABR and ABSR Information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron#show ip ospf vrf green border-routers
      router ID       router type next hop router outgoing interface Area
1     1.2.10.2        ASBR        192.168.201.2   6/3                1
1     10.5.1.3        ASBR        192.168.201.2   6/3                1
```

Syntax: show ip ospf vrf <*vrf-name*> **border-routers** <*router-id*>

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF ABR and ABSR information for.

The < *router-id*> parameter specifies the display of OSPF ABR and ABSR information for the router with the specified router ID.

Displaying general OSPF configuration information for a VRF

To display OSPF ABR and ABSR Information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron#show ip ospf vrf green config
Router OSPF: Enabled
Redistribution: Enabled
Default OSPF Metric: 10
OSPF Auto-cost Reference Bandwidth: Disabled

OSPF Redistribution Metric: Type2

OSPF External LSA Limit: 14447047

OSPF Database Overflow Interval: 0

RFC 1583 Compatibility: Enabled

Router id: 192.168.201.1
Interface State Change Trap:                      Enabled
Virtual Interface State Change Trap:              Enabled
Neighbor State Change Trap:                       Enabled
Virtual Neighbor State Change Trap:               Enabled
Interface Configuration Error Trap:               Enabled
Virtual Interface Configuration Error Trap:       Enabled
Interface Authentication Failure Trap:            Enabled
Virtual Interface Authentication Failure Trap:    Enabled
Interface Receive Bad Packet Trap:                Enabled
Virtual Interface Receive Bad Packet Trap:        Enabled
Interface Retransmit Packet Trap:                 Disabled
Virtual Interface Retransmit Packet Trap:         Disabled
Originate LSA Trap:                               Disabled
Originate MaxAge LSA Trap:                        Disabled
Link State Database Overflow Trap:                Disabled
Link State Database Approaching Overflow Trap:    Disabled

OSPF Area currently defined:
Area-ID          Area-Type Cost
0                normal    0
1                normal    0
```

Syntax: show ip ospf vrf <*vrf-name*> **config**

The <*vrf-name*> parameter specifies the VRF that you want to display general OSPF configuration information for.

Displaying OSPF external link state information for a VRF

To display OSPF External Link State Information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron#show ip ospf vrf green database external-link-state
Index Aging  LS ID           Router          Netmask  Metric    Flag
1     491    10.30.1.6       10.5.1.3        ffffffff 00000001  0000
2     1005   10.40.1.30      192.168.201.1   ffffffff 8000000a  0000
3     765    10.60.1.10      192.168.201.1   ffffffff 8000000a  0000
4     1005   10.40.1.9       192.168.201.1   ffffffff 8000000a  0000
5     491    10.30.1.19      10.5.1.3        ffffffff 00000001  0000
6     765    10.60.1.23      192.168.201.1   ffffffff 8000000a  0000
7     1005   10.40.1.22      192.168.201.1   ffffffff 8000000a  0000
8     765    10.60.1.2       192.168.201.1   ffffffff 8000000a  0000
9     1005   10.40.1.1       192.168.201.1   ffffffff 8000000a  0000
10    491    10.30.1.11      10.5.1.3        ffffffff 00000001  0000
11    765    10.60.1.15      192.168.201.1   ffffffff 8000000a  0000
12    1005   10.40.1.14      192.168.201.1   ffffffff 8000000a  0000
13    491    10.30.1.24      10.5.1.3        ffffffff 00000001  0000
14    491    10.30.1.3       10.5.1.3        ffffffff 00000001  0000
```

Syntax: show ip ospf vrf <*vrf-name*> **database external-link-state [advertise** <*num*>**] | [extensive] | [link-state-id** <*ip-addr*>**] | [router-id** <*ip-addr*>**] | [sequence-number** <*num (Hex)* >**] | [status** <*num*>**]**

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF external link state information for.

The advertise <*num*> parameter displays the data in the specified LSA packet. The <*num*> parameter identifies the LSA packet by its position in the router's External LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf vrf** <*vrf-name*> **external-link-state** command to display the table.

The extensive option displays the data in the LSAs in decrypted format.

The link-state-id <*ip-addr*> parameter displays the External LSAs for the LSA source specified by <*IP-addr*>.

The router-id <*ip-addr*> parameter shows the External LSAs for the specified OSPF router.

The status <*num*> option shows status information.

The sequence-number <*num (Hex)*> parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

Displaying OSPF link state information for a VRF

To display OSPF Link State Information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron#show ip ospf vrf green database link-state
Index Area ID         Type LS ID           Adv Rtr         Seq(Hex) Age  Cksum
1     0               Summ 1.2.10.2        192.168.201.1   8000001b 1145 0x03fb
2     0               Summ 192.168.201.0   192.168.201.1   8000001b 1145 0x4d8d
3     0               Summ 10.8.1.0        192.168.201.1   8000001b 905  0xadc5
4     0               Summ 10.5.1.0        192.168.201.1   8000001b 1145 0xea12
5     0               ASBR 1.2.10.2        192.168.201.1   8000001b 1145 0xf409
6     0               ASBR 10.5.1.3        192.168.201.1   8000001b 1145 0xbe3a
7     1               Rtr  192.168.201.1   192.168.201.1   80000088 1145 0xf304
8     1               Rtr  1.2.10.2        1.2.10.2        800000eb 581  0x503d
9     1               Rtr  10.5.1.3        10.5.1.3        8000005e 1470 0xf8b0
10    1               Net  192.168.201.1   192.168.201.1   8000001f 1145 0xb5da
11    1               Net  10.5.1.1        1.2.10.2        8000004e 1792 0x0fbb
12    1               Summ 10.8.1.0        192.168.201.1   8000001b 905  0xadc5
```

Syntax: **show ip ospf vrf** <*vrf-name*> **database link-state [advertise** <*num*>**] | [asbr] | [extensive] | [link-state-id** <*ip-addr*>**] |[network] | [nssa] | [opaque-area] | [router] | [router-id** <*ip-addr*>**] | [sequence-number** <*num (Hex)*>**] | [status** <*num*>**] | [summary]**

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF link state information for.

The advertise <*num*> parameter displays the hexadecimal data in the specified LSA packet. The <*num*> parameter identifies the LSA packet by its position in the router's External LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf vrf** <*vrf-name*> **external-link-state** command to display the table.

The asbr option shows ASBR information.

The extensive option displays the LSAs in decrypted format.

The link-state-id <*ip-addr*> parameter displays the External LSAs for the LSA source specified by <*IP-addr*>.

The network option shows network information.

The nssa option shows network information.

The opaque-area option shows information for opaque areas.

The router-id <*ip-addr*> parameter shows the External LSAs for the specified OSPF router.

The sequence-number <*num (Hex)*> parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

The status <*num*> option shows status information.

The summary option shows summary information.

Displaying OSPF interface information

To display OSPF interface information for a specified VRF, enter the following command at any CLI level.

```
NetIron# show ip ospf vrf green interface
ethernet 6/3,OSPF enabled
     IP Address 192.168.201.1, Area 1
     OSPF state DR, Pri 1, Cost 1, Options 2, Type broadcast Events 3
     Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
     DR:  Router ID 192.168.201.1     Interface Address 192.168.201.1
     BDR: Router ID 1.2.10.2          Interface Address 192.168.201.2
     Neighbor Count = 1, Adjacent Neighbor Count= 1
     Neighbor:         192.168.201.2
     Authentication-Key:None
     MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

Syntax: show ip ospf vrf <*vrf-name*> **interface** [<*ip-addr*>]

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF interface information for.

The <*ip-addr*> parameter displays the OSPF interface information for the specified IP address.

Displaying OSPF neighbor information for a VRF

To display OSPF neighbor information for a specified VRF, enter the following command at any CLI level.

```
NetIron# show ip ospf vrf green neighbor

Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
6/3   192.168.201.1   1   FULL/BDR   192.168.201.2   1.2.10.2        6  2   0
```

Syntax: show ip ospf vrf <*vrf-name*> **neighbor [router-id** <*ip-addr*>**] | [**<*num*>**] | [detail]**

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF neighbor information for.

The router-id <*ip-addr*> parameter displays only the neighbor entries for the specified router.

The <*num*> parameter displays only the entry in the specified index position in the neighbor table. For example, if you enter "1", only the first entry in the table is displayed.

The detail parameter displays detailed information about the neighbor routers.

Displaying the routes that have been redistributed into OSPF

You can display the routes that have been redistributed into OSPF for a VRF. To display the redistributed routes, enter the following command at any level of the CLI.

```
NetIron# show ip ospf vrf green redistribute route
  10.6.1.0 255.255.255.0 bgp
  10.8.1.0 255.255.255.0 bgp
  10.40.1.1 255.255.255.255 bgp
  10.40.1.2 255.255.255.255 bgp
```

In this example, four routes have been redistributed from BGP routes

Syntax: show ip ospf vrf <*vrf-name*> **redistribute route**

The <*vrf-name*> parameter specifies the VRF that you want to display routes redistributed into OSPF for.

Displaying OSPF route information for a VRF

To display the OSPF route information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron# show ip ospf vrf green routes
OSPF Area 0x00000001 ASBR Routes 2:
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      1.2.10.2        255.255.255.255 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      1.2.10.2        1.2.10.2        Asbr      Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     6/3       192.168.201.2   OSPF      00 00
```

In this example, four routes have been redistributed from BGP routes

Syntax: show ip ospf vrf <*vrf-name*> **routes** [<*ip-addr*>]

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF routes for.

The <*ip-addr*> parameter specifies a destination IP address. If you use this parameter, only the route entries for that destination are shown.

Displaying OSPF sham links

To display the OSPF sham links information for a VRF, enter the **show ip ospf vrf** <*vrf-name*> **sham-links** command at any level of the CLI, as in the following example.

```
NetIron# show ip ospf vrf CustomerA sham-links
Sham Link in OSPF instance CustomerA to 10.1.1.2 is UP, Established over lsp(LDP)
 Area 1 source address 10.1.1.1
   Link cost 1 Transmit Delay is 1 sec, State ptpt
   Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Adjacency State UP, number of interface events 417
```

Syntax: show ip ospf vrf <*vrf-name*> **sham-links**

The <*vrf-name*> variable identifies the VRF for which you want to display OSPF sham links information.

Displaying OSPF trap status for a VRF

To display the state (enabled or disabled) of the OSPF traps for a specified VRF, enter the following command at any CLI level.

```
NetIron# R3-2547#show ip ospf vrf green trap
Interface State Change Trap:                      Enabled
Virtual Interface State Change Trap:              Enabled
Neighbor State Change Trap:                       Enabled
Virtual Neighbor State Change Trap:               Enabled
Interface Configuration Error Trap:               Enabled
Virtual Interface Configuration Error Trap:       Enabled
Interface Authentication Failure Trap:            Enabled
Virtual Interface Authentication Failure Trap:    Enabled
Interface Receive Bad Packet Trap:                Enabled
Virtual Interface Receive Bad Packet Trap:        Enabled
Interface Retransmit Packet Trap:                 Disabled
Virtual Interface Retransmit Packet Trap:         Disabled
Originate LSA Trap:                               Disabled
Originate MaxAge LSA Trap:                        Disabled
Link State Database Overflow Trap:                Disabled
Link State Database Approaching Overflow Trap:    Disabled
```

Syntax: show ip ospf vrf <*vrf-name*> **trap**

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF trap status for.

Displaying OSPF virtual links for a VRF

To display the OSPF virtual links information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron# show ip ospf vrf green virtual-links
No ospf virtual-link entries available
```

Syntax: show ip ospf vrf <*vrf-name*> **virtual-link** [<*num*>]

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF virtual links information for.

The <*num*> parameter displays the table beginning at the specified entry number.

Displaying OSPF virtual neighbor information for a VRF

To display the OSPF virtual neighbor information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron# R3-2547#show ip ospf vrf green virtual-neighbor
```

Syntax: show ip ospf vrf <*vrf-name*> **virtual-neighbor** [<*num*>]

The <*vrf-name*> parameter specifies the VRF that you want to display OSPF virtual neighbor information for.

The <*num*> parameter displays the table beginning at the specified entry number.

Displaying IP extcommunity list information

To display the IP Extcommunity information, enter the following command at any level of the CLI.

```
NetIron#show ip extcommunity-list
ip extcommunity access list 20:
  permit RT 100:1
```

Syntax: show ip extcommunity-list

For information about the fields, refer to the following.

TABLE 298 Output of show IP extcommunity list

This field...	Displays...
ip extcommunity access list	The contents of all extended community lists on the router.

Displaying the IP static route table for a VRF

To display the IP static route table for a VRF, enter the following command at any level of the CLI.

```
NetIron # show ip static route vrf green
IP Static Routing Table - entries:
    IP Prefix         Next Hop      Interface     Dis/Met/Tag   Name
    10.22.66.0/24     10.22.66.0    -             1/1/0         green
```

Syntax: show ip static route vrf <*vrf-name*>

The <*vrf-name*> parameter specifies the VRF that you want to display the static route table for.

Show run displays the entire name of the static IP route. The **show ip static route** command displays an asterisk (*) after the first twelve characters if the assigned name is thirteen characters or more. The **show ipv6 static route** command displays an asterisk after the first two characters if the assigned name is three characters or more.

Displaying the static ARP table for a VRF

To display the static ARP table for a VRF, enter the following command at any level of the CLI.

```
NetIron# show ip static-arp vrf green
Static ARP table size: 2048, configurable from 2048 to 4096
  Index   IP Address          MAC Address         Port
    1      207.95.6.111       0800.093b.d210       1/1
    3      207.95.6.123       0800.093b.d211       1/1
```

Syntax: show ip static-arp vrf <*vrf-name*>

The <*vrf-name*> parameter specifies the VRF that you want to display the static ARP table for.

To clear the static ARP table in a VRF, enter the following command.

```
PE1# clear arp vrf blue
```

Syntax: clear arp vrf <*vrf-name*>

Displaying TCP connections for a VRF

The **show ip tcp vrf connections** command displays information about each TCP connection on the VRF, including the local IP address, local port number, remote IP address, remote port number and the state of the connection. For example.

```
NetIron# show ip tcp vrf green connections
Local IP address:port <-> Remote IP address:port TCP state    (hdl itc cln pdn)
0.0.0.0:179 <-> 0.0.0.0:0 LISTEN (000100bf: 13, 0, 0)
Total 1 TCP connections
```

Syntax: show ip tcp vrf <*vrf-name*> **connections**

The <*vrf-name*> parameter specifies the VRF that you want to display TCP connections for.

Displaying MPLS statistics for a VRF

To display MPLS statistics on a per-interface basis for a specified VRF, enter the following command at any level of the CLI.

```
NetIron# show mpls statistics vrf green
VRF Name       In-Port(s)     Endpt Out-Pkt     Tnl Out-Pkt
green          e3/1           0                 0
               e3/2           0                 0
               e3/3           0                 0
               e3/4           0                 0
               e6/1           0                 0
               e6/2           4367535952        0
               e6/3           0                 4366414365
               e6/4           0                 0
```

Syntax: show mpls statistics vrf <*vrf-name*>

The <*vrf-name*> parameter specifies the VRF that you want to display MPLS statistics for.

For information about the fields in this display, refer the following.

TABLE 299 Output from the show MPLS statistics VRF command

This field...	Displays...
VRF Name	The name of the VRF MPLS statistics are being collected for.
In-Ports	The port where the traffic is received.
Endpt Out-Pkt	The number of packets transmitted out of local endpoints.
Tnl Out-Pkt	The number of packets transmitted out of lsp tunnels.

To clear the MPLS statistics counters.

```
NetIron# clear mpls statistics
```

Syntax: clear mpls statistics [label | tunnel | vpls | vrf]

The **label** parameter clears in-label statistics.

The **tunnel** parameter clears MPLS tunnel statistics.

The **vpls** parameter clears VPLS statistics.

The **vrf** parameter clears vrf statistics.

Displaying IP route information for a VRF

Display IP route Information for a specified VRF by entering the following command.

```
NetIron#show ip route vrf yellow
Total number of IP routes: 2
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
       Destination        Gateway         Port        Cost       Type
1      8.8.8.8/32         DIRECT          loopback 1  0/0        D
2      9.9.9.8/32         20.0.0.1        lsp to1     200/0      B
```

Syntax: show ip route vrf <*vrf-name*> [<*num*>] | [<*ip-addr*>] | **[bgp]** | **[connected]** |**[isis]** | **[ospf]** | **[rip]** | **[static]** | **[tags]**

The <*vrf-name*> parameter specifies the VRF that you want to display IP route information for.

Displaying RIP information for a VRF

To display RIP Information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron#show ip rip vrf black
RIP Summary
  Default port 520
      Administrative distance is 120
      Updates every 30 seconds, expire after 180
      Holddown lasts 180 seconds, garbage collect after 120
      Last broadcast  27, Next Update 29
      Need trigger update  0, Next trigger broadcast 3
      Minimum update  interval 25, Max update Interval 5
      Split horizon is on; poison reverse is off
      Import metric 1
      Prefix List, Inbound : Not set
      Prefix List, Outbound : Not set
      Route-map, Inbound : Not set
```

Syntax: show ip rip vrf <*vrf-name*> **[interface [ethernet** <*slot/port*>**]] | [route** <*ipaddress*>**]**

The <*vrf-name*> parameter specifies the VRF that you want to display IP route information for.

To clear all RIP routes from a specified VRF, enter the following command.

```
PE1# clear ip rip routes vrf blue
```

Syntax: clear ip rip routes vrf <*vrf-name*>

To clear all local RIP routes from a specified VRF, enter the following command.

```
PE1# clear ip rip local routes vrf blue
```

Syntax: clear ip rip local routes vrf <*vrf-name*>

BGP or MPLS VPN sample configurations

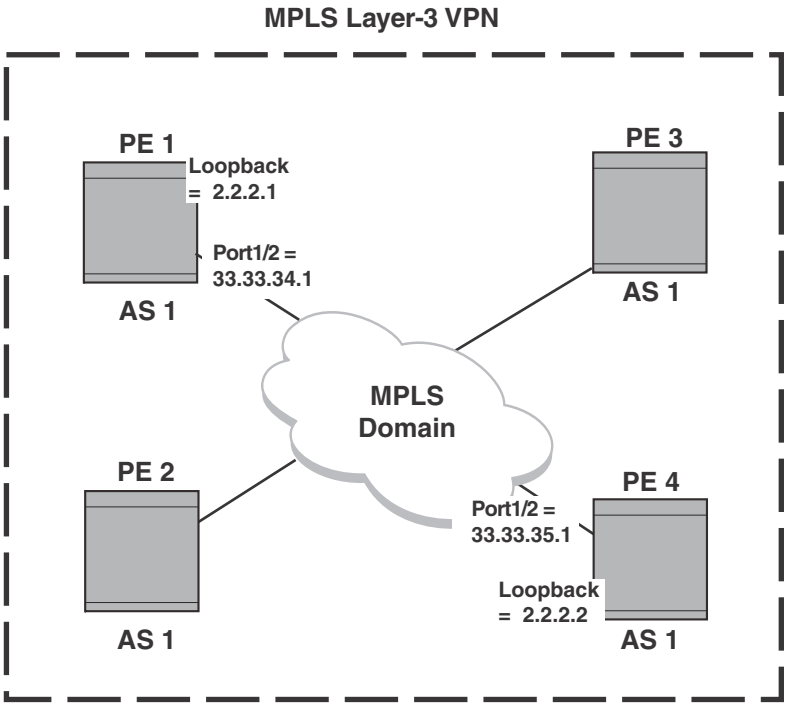
This section presents examples of typical MPLS configurations. The following sample configurations are presented:

- "Basic configuration example for IBGP on the PEs"
- "EBGP for route exchange"
- "Static routes for route exchange"
- "RIP for route exchange"
- "OSPF for route exchange"
- "Cooperative route filtering"
- "Using an IP extcommunity variable with route map"
- "Autonomous system number override"
- "Setting an LSP for each VRF on a PE"
- "OSPF sham links"

Basic configuration example for IBGP on the PEs

PE routers use IBGP to exchange VRF routes. As in all BGP configurations, this is accomplished by configuring BGP neighbors where you want to exchange routes. If the neighbors are configured in the same AS, it will be an IBGP configuration. In addition, since MPLS LSPs are made between router loopback addresses, the [update-source loopback] parameters must be used. The example in Figure 206 shows two PE routers (PE 1 and PE 4) that are configured as BGP neighbors.

FIGURE 206 IBGP example



To configure IBGP on a Provider Edge router (PE) of a BGP or MPLS VPN network, you must perform the configuration steps listed beolw.

1. "Assigning an AS number to a PE"
2. "Assigning a loopback interface"
3. "Configuring an IBGP neighbor on a PE"

Assigning an AS number to a PE

In the IBGP configuration used in a BGP or MPLS VPN, all PEs are configured with the same AS number. To assign the local AS number 1 to the PE 1 router as shown in Figure 206, enter the following commands.

```
PE1(config)#router bgp
PE1(config-bgp)# local-as 1
```

Assigning a loopback interface

A loopback interface is used as the termination for address for BGP sessions. This allows BGP to stay up even when the outbound interface is down as long as an alternate path is available. To install the loopback interface on PE 1 as shown in Figure 206, enter the following commands.

```
PE1(config)#interface loopback 1
PE1(config-lbif-1)# ip address 2.2.2.1/32
```

Configuring an IBGP neighbor on a PE

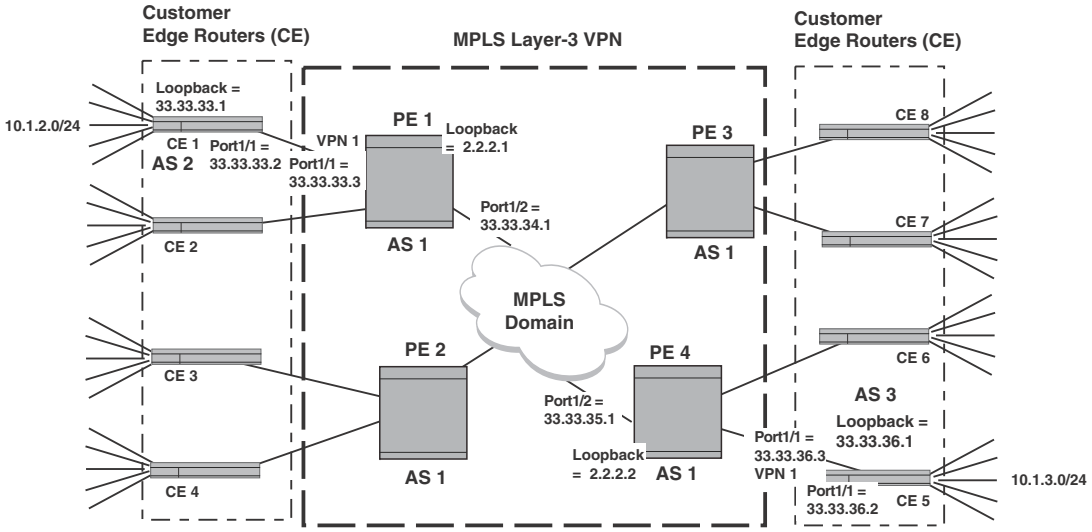
Other PEs that you want to exchange IBGP routes with must be configured as BGP neighbors. In addition, the neighbor must be set to enable the BGP to update the loopback address. To assign an IBGP neighbor with the IP address 33.33.35.1, a remote AS number of 1, and an update-source to loopback 1 of the PE 1 router shown in Figure 206, enter the following commands.

```
PE1(config-bgp)# neighbor 2.2.2.2 remote-as 1
PE1(config-bgp)# neighbor 2.2.2.2 update-source loopback 1
PE1(config-bgp)# address-family vpnv4 unicast
PE1(config-bgp-vpnv4u)# neighbor 2.2.2.2 activate
```

EBGP for route exchange

EBGP can be used to exchange routes for CE routers to PE routers. In this situation, a BGP neighbor must be configured on CE and PE routers. In the example shown in Figure 207, the CE 1 router is configured to exchange routes with the PE 1 router and the CE 5 router is configured to exchange routes with the PE 4 router.

FIGURE 207 EBGP to CE network example



To configure EBGP to exchange routes between PE routers and CE routers, you must perform the configuration steps listed below.

1. "Configuring EBGP on a CE router".
2. "Configuring EBGP on a PE router".

Configuring EBGP on a CE router

To allow route exchange between a CE router and its associated PE router, BGP must be enabled on the CE router and the associated PE router must be configured as a BGP neighbor. To enable BGP on CE 1 and assign PE 1 as a BGP neighbor in the network shown in Figure 207, enter the following commands.

```
CE1(config)# router bgp
CE1(config-bgp)# local-as 2
CE1(config-bgp)# neighbor 33.33.33.3 remote-as 1
```

Configuring EBGP on a PE router

To allow route exchange between a VRF on a PE router and its associated CE router, BGP must be enabled on the appropriate VRF of the PE router and the associated CE router must be configured as a BGP neighbor. To assign CE 1 as a BGP neighbor to the VRF VPN1 on PE 1 in the network shown in Figure 207, enter the following commands.

```
PE1(config-bgp)# address-family ipv4 unicast vrf VPN1
PE1(config-bgp-ipv4u-vrf)# neighbor 33.33.33.2 remote-as 2
```

EBGP to CE network example

In the example shown in Figure 207, the network is configured to use EBGP to forward routes between the networks attached to the CE routers and the PE routers. IBGP is used to forward routes between the PE routers and an LSP tunnel is configured across the MPLS Domain. Figure 207 contains all of the network addresses and AS numbers required to perform this configuration. The configurations are shown for only the CE 1, CE 5, PE 1 and PE 5 routers, which demonstrates what is required to make VPN1 work. The configurations for VPN2, VPN3, and VPN 4 would essentially be the same.

CE 1 configuration

This configuration example describes what is required to operate the CE 1 router in Figure 207. In this example, a static route is configured between the external network (10.1.2.0/24) and the loopback interface. EBGP is configured between CE 1 and PE 1, and the static route is redistributed through this connection.

```
CE1(config)# ip route 10.1.2.0/24 33.33.33.1
CE1(config)# router bgp
CE1(config-bgp)# local-as 2
CE1(config-bgp)# neighbor 33.33.33.3 remote-as 1
CE1(config-bgp)# redistribute static
CE1(config-bgp)# exit

CE1(config)# interface ethernet 1/1
CE1(config-if-e10000-1/1)# ip address 33.33.33.2/24
CE1(config-if-e10000-1/1)# exit
```

CE 5 configuration

This configuration example describes what is required to operate the CE 5 router in Figure 207. In this example, a static route is configured between the external network (10.1.3.0/24) and the loopback interface. EBGP is configured between CE 5 and PE 4, and the static route is redistributed through this connection.

```
CE5(config)#interface loopback 1
CE5(config-lbif-1)# ip address 33.33.36.1/32
CE5(config-lbif-1)# exit

CE5(config)# ip route 10.1.3.0/24 33.33.36.1
CE5(config)# router bgp
CE5(config)# local-as 3
CE5(config-bgp)# neighbor 33.33.36.3 remote-as 1
CE5(config-bgp)# redistribute static
CE5(config-bgp)# exit

CE5(config)#interface ethernet 1/1
CE5(config-if-e10000-1/1)# ip address 33.33.36.2/24
CE5(config-if-e10000-1/1)# exit
```

PE 1 configuration

This configuration example describes what is required to operate the PE 1 router in Figure 207. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 1. EBGP is configured between VPN1 and CE 1. IBGP with extended community attributes is configured between PE 1 and PE 4.

The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signalled LSP named “tunnel1” to PE 4.

```
PE1(config)#interface loopback 1
PE1(config-lbif-1)# ip address 2.2.2.1/32
PE1(config-lbif-1)# ip ospf area 0
PE1(config-lbif-1)# exit

PE1(config)#ip vrf VPN1
PE1(config-ip-vrf-vpn1)#rd 1:1
PE1(config-ip-vrf-vpn1)#route-target export 100:1
PE1(config-ip-vrf-vpn1)#route-target import 100:2
PE1(config-ip-vrf-vpn1)# exit-vrf

PE1(config)# router bgp
PE1(config-bgp)# local-as 1
PE1(config-bgp)# neighbor 2.2.2.2 remote-as 1
PE1(config-bgp)# neighbor 2.2.2.2 update-source loopback 1
PE1(config-bgp)# address-family vpnv4 unicast
PE1(config-bgp-vpnv4u)# neighbor 2.2.2.2 activate
PE1(config-bgp-vpnv4u)# exit

PE1(config-bgp)# address-family ipv4 unicast vrf VPN1
PE1(config-bgp-ipv4u-vrf)# neighbor 33.33.33.2 remote-as 2
PE1(config-bgp-ipv4u-vrf)# exit

PE1(config)# router ospf
PE1(config-ospf-router)# area 0
PE1(config-ospf-router)# exit

PE1(config)#interface ethernet 1/2
PE1(config-if-e10000-1/2)# ip address 33.33.34.1/24
PE1(config-if-e10000-1/2)# ip ospf area 0
PE1(config-if-e10000-1/2)# exit

PE1(config)# router mpls
PE1(config-mpls)# policy
PE1(config-mpls-policy)# traffic-engineering ospf
PE1(config-mpls)# mpls-interface eth 1/2
PE1(config-mpls)# lsp tunnel1
PE1(config-mpls-lsp-tunnel1)# to 2.2.2.2
PE1(config-mpls-lsp-tunnel1)# enable
PE1(config-mpls-lsp-tunnel1)# exit

PE1(config)#interface ethernet 1/1
PE1(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE1(config-if-e10000-1/1)# ip address 33.33.33.3/24
```

PE 4 configuration

This configuration example describes what is required to operate the PE 2 router in Figure 207. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 5. EBGP is configured between VPN1 and CE 5. IBGP with extended community attributes is configured between PE 4 and PE 1.

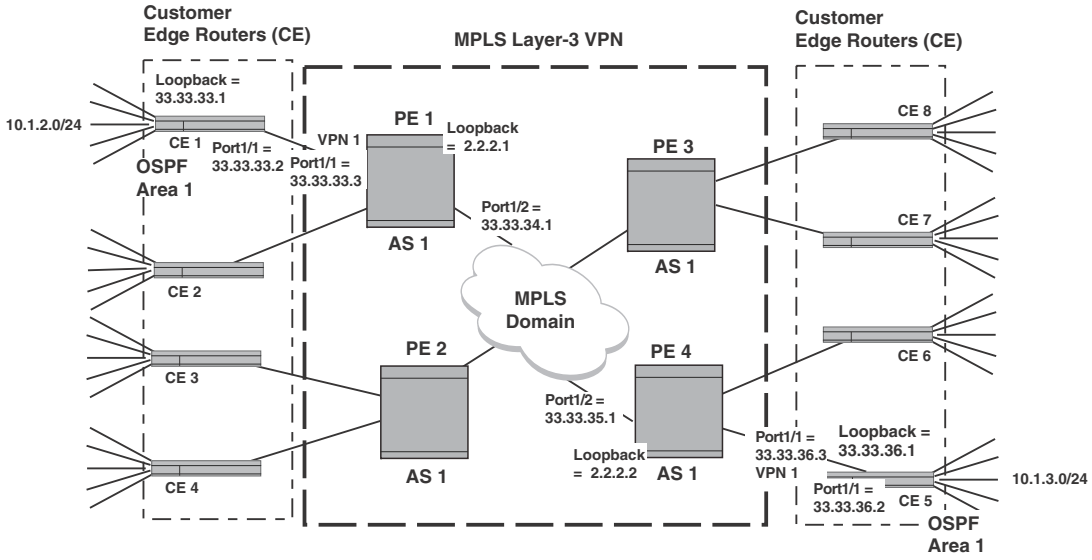
The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signalled LSP named “tunnel1” to PE 1.

```
PE4(config)#interface loopback 1
PE4(config-lbif-1)# ip address 2.2.2.2/32
PE4(config-lbif-1)# ip ospf area 0
PE4(config-lbif-1)# exit

PE4(config)#ip vrf VPN1
PE4(config-vrf-vpn1)#rd 1:2
PE4(config-vrf-vpn1)#route-target export 100:1
PE4(config-vrf-vpn1)#route-target import 100:2
PE4(config-vrf-vpn1)# exit-vrf

PE4(config)# router bgp
PE4(config-bgp)# local-as 1
PE4(config-bgp)# neighbor 2.2.2.1 remote-as 1
PE4(config-bgp)# neighbor 2.2.2.1 update-source loopback 1
PE4(config-bgp)# address-family vpnv4 unicast
PE1(config-bgp-vpnv4u)# neighbor 2.2.2.1 activate
PE4(config-bgp)# address-family ipv4 unicast vrf VPN1
PE4(config-bgp-ipv4u-vrf)# neighbor 33.33.36.2 remote-as 2
PE4(config-bgp-ipv4u-vrf)# exit

PE4(config)# router ospf
PE4(config-ospf-router)# area 0
PE4(config-ospf-router)# exit

PE4(config)#interface ethernet 1/2
PE4(config-if-e10000-1/2)# ip address 33.33.35.1/24
PE4(config-if-e10000-1/2)# ip ospf area 0
PE4(config-if-e10000-1/2)# exit

PE4(config)# router mpls
PE4(config-mpls)# policy
PE4(config-mpls-policy)# traffic-engineering ospf
PE4(config-mpls)# mpls-interface eth 1/2
PE4(config-mpls)# lsp tunnel1
PE4(config-mpls-lsp-tunnel1)# to 2.2.2.1
PE4(config-mpls-lsp-tunnel1)# enable
PE4(config-mpls-lsp-tunnel1)# exit

PE4(config)#interface ethernet 1/1
PE4(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE4(config-if-e10000-1/1)# ip address 33.33.36.3/24
PE4(config-if-e10000-1/1)# exit
```

Static routes for route exchange

Static routes can be used to exchange routes between CE routers and PE routers. In this situation, a default static route must be configured on a CE router to its associated PE router. A static route must be configured between the PE router and the network (or networks) that the PE wants to advertise as available through a VRF.

FIGURE 208 Static route to CE network example



To configure static routes to exchange routes between PE routers and CE routers, you must perform the configuration steps listed below.

1. "Configuring a static default route on a CE router"
2. "Configuring a static default route on a PE router"

Configuring a static default route on a CE router

To allow route exchange between a CE router and its associated PE router, a static default route must be created to the interface on the associated PE router where the VPN is enabled. In Figure 208, the PE 1 router has the VRF "VPN1" enabled on port 1/1, which has the IP address 33.33.33.3. To create a default static route from CE 1 to this interface on PE 1, enter the following command.

```
CE1(config)# ip route 0.0.0.0 33.33.33.3
```

Configuring a static default route on a PE router

To allow route exchange between a PE router and its associated CE router, a static route must be created to the route that you want to provide access to with a next hop consisting of the IP address of the interface that is connected to the VRF. In Figure 208, the IP address of the connected port on the CE router is 33.33.33.2, and the address on the CE that will be provided access from the PE's VRF is 10.1.2.0/24. To create a static route from PE 1 to CE 1, enter the following command.

```
PE1(config)# ip route vrf VPN1 10.1.2.0/24 33.33.33.2
```

Static route to CE example

In this example, the network shown in Figure 208 is configured for a default static route to forward routes between the networks attached to the CE routers and the PE routers. IBGP is used to forward routes between the PE routers and an LSP tunnel is configured across the MPLS domain. Figure 208 contains all of the network addresses and AS numbers required to perform this configuration. The configurations are shown for only the CE 1, CE 5, PE 1 and PE 5 routers which demonstrates what is required to make VPN1 work. The configurations for VPN2, VPN3, and VPN 4 would essentially be the same.

CE 1 configuration

This configuration example describes what is required to operate the CE 1 router in Figure 208. In this example, a default static route is configured between the CE 1 router and the attached interface of PE 1.

```
CE1(config)# ip route 0.0.0.0/0 33.33.33.3
CE1(config)#interface ethernet 1/1
CE1(config-if-e10000-1/1)# ip address 33.33.33.2
```

CE 5 configuration

This configuration example describes what is required to operate the CE 5 router in Figure 208. In this example, a default static route is configured between the CE 5 router and the attached interface of PE 4.

```
CE5(config)# ip route 0.0.0.0/0 33.33.36.3
CE5(config)#interface ethernet 1/1
CE5(config-if-e10000-1/1)# ip address 33.33.34.2
```

PE 1 configuration

This configuration example describes what is required to operate the PE 1 router in Figure 208. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 1. A static route is configured between this router and the network connected to CE 1. IBGP with extended community attributes is configured between PE 1 and PE 4.

The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signaled LSP named “tunnel1” to PE 4.

```
PE1(config)#interface loopback 1
PE1(config-lbif-1)# ip address 2.2.2.1/24
PE1(config-lbif-1)# ip ospf area 0
PE1(config-lbif-1)# exit

PE1(config)#ip vrf VPN1
PE1(config-vrf-vpn1)# rd 1:1
PE1(config-vrf-vpn1)# route-target export 100:1
PE1(config-vrf-vpn1)# route-target import 100:2
PE1(config-vrf-vpn1)# exit-vrf

PE1(config)# ip route vrf VPN1 10.1.2.0/24 33.33.33.2
PE1(config)# router bgp
PE1(config-bgp)# local-as 1
```

```
PE1(config-bgp)# neighbor 2.2.2.2 remote-as 1
PE1(config-bgp)# neighbor 2.2.2.2 update-source loopback 1
PE1(config-bgp)# address-family vpnv4 unicast
PE1(config-bgp-vpnv4u)# neighbor 2.2.2.2 activate
PE1(config-bgp)# address-family ipv4 unicast vrf VPN1
PE1(config-bgp-ipv4u-vrf)# redistribute static
PE1(config-bgp-ipv4u-vrf)# exit

PE1(config)# router ospf
PE1(config-ospf-router)# area 0
PE1(config-ospf-router)# exit

PE1(config)# interface ethernet 1/2
PE1(config-if-e10000-1/2)# ip address 33.33.34.1/24
PE1(config-if-e10000-1/2)# ip ospf area 0
PE1(config-if-e10000-1/2)# exit

PE1(config)# router mpls
PE1(config-mpls)# policy
PE1(config-mpls-policy)# traffic-engineering ospf
PE1(config-mpls)# mpls-interface eth 1/2
PE1(config-mpls)# lsp tunnel1
PE1(config-mpls-lsp-tunnel1)# to 2.2.2.2
PE1(config-mpls-lsp-tunnel1)# enable
PE1(config-mpls-lsp-tunnel1)# exit

PE1(config)#interface ethernet 1/1
PE1(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE1(config-if-e10000-1/1)# ip address 33.33.33.3/24
PE1(config-if-e10000-1/1)#
```

PE 4 configuration

This configuration example describes what is required to operate the PE 4 router in Figure 208. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 5. A static route is configured between this router and the network connected to CE 5.

IBGP with extended community attributes is configured between PE 1 and PE 4. The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signalled LSP named “tunnel1” to PE 1.

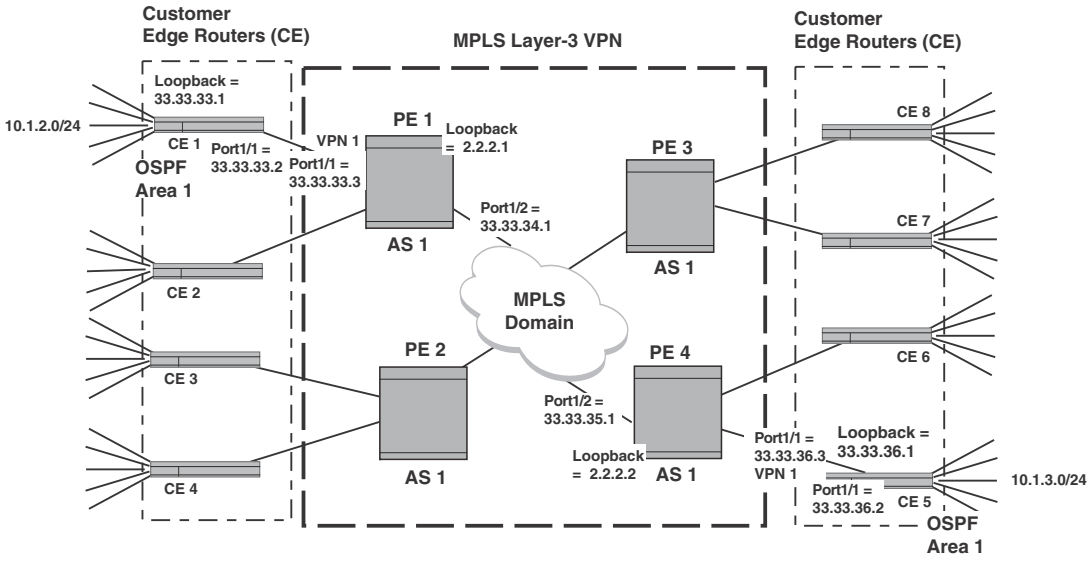
```
PE4(config)#interface loopback 1
PE4(config-lbif-1)# ip address 2.2.2.2/32
PE4(config-lbif-1)# ip ospf area 0
PE4(config-lbif-1)# exit

PE4(config)#ip vrf VPN1
PE4(config-vrf-vpn1)# rd 1:2
PE4(config-vrf-vpn1)# route-target export 100:1
PE4(config-vrf-vpn1)# route-target import 100:2
PE4(config-vrf-vpn1)# exit-vrf

PE4(config)# ip route vrf VPN1 10.1.2.0/24 33.33.36.2
PE4(config)# router bgp
PE4(config-bgp)# local-as 1
PE4(config-bgp)# neighbor 2.2.2.1 remote-as 1
PE4(config-bgp)# neighbor 2.2.2.1 update-source loopback 1
```

```
PE4(config-bgp)# address-family vpnv4 unicast
PE4(config-bgp-vpnv4u)# neighbor 2.2.2.1 activate
PE4(config-bgp)# address-family ipv4 unicast vrf VPN1
PE4(config-bgp-ipv4u-vrf)# redistribute static
PE4(config-bgp-ipv4u-vrf)# exit

PE4(config)# router ospf
PE4(config-ospf-router)# area 0
PE4(config-ospf-router)# exit

PE4(config)#interface ethernet 1/2
PE4(config-if-e10000-1/2)# ip address 33.33.35.1/24
PE4(config-if-e10000-1/2)# ip ospf area 0
PE4(config-if-e10000-1/2)# exit

PE4(config)# router mpls
PE4(config-mpls)# policy
PE4(config-mpls-policy)# traffic-engineering ospf
PE4(config-mpls)# mpls-interface eth 1/2
PE4(config-mpls)# lsp tunnel1
PE4(config-mpls-lsp-tunnel1)# to 2.2.2.1
PE4(config-mpls-lsp-tunnel1)# enable
PE4(config-mpls-lsp-tunnel1)# exit

PE4(config)# interface ethernet 1/1
PE4(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE4(config-if-e10000-1/1)# ip address 33.33.36.3/24
PE4(config-if-e10000-1/1)# exit
```

RIP for route exchange

RIP can be used to exchange routes between CE routers and PE routers. In this situation, RIP must be enabled on the CE router and enabled on the interface that is connected to the interface of the PE that is associated with the VRF that you want to advertise RIP routes on. On the PE router, the VRF must be enabled to redistribute RIP routes, and RIP must be enabled for the VRF and configured to redistribute routes from BGP in the VRF. Figure 209 provides an example of a network where RIP is used to exchange routes between CE routers and PE routers.

FIGURE 209 RIP to CE network example



To configure RIP to exchange routes between PE routers and CE routers, you must perform the configuration steps listed below.

1. "Configuring RIP on the CE router"
2. "Enabling RIP on the CE router' interface"
3. "Configuring the VRF on the PE router to redistribute RIP routes"
4. "Configuring RIP on the PE router to redistribute BGP-VPNv4 routes"
5. "Enabling RIP on the PE router interface"

Configuring RIP on the CE router

To allow RIP route exchange between a CE router and its associated PE router, RIP must be enabled on the CE router. To configure RIP on the CE 1 router in Figure 209 and enable it to redistribute static routes through RIP, enter the following commands.

```
CE1(config)# router rip
CE1(config-rip-router)# redistribute static
```

Enabling RIP on the CE router' interface

To allow RIP route exchange between a CE router and its associated PE router, RIP must be enabled on the interface that connects to the VRF-enabled interface of its associated PE router. To configure RIP on the interface of the CE 1 router in Figure 209 that is connected to the VRF VPN1 associated interface on PE 1, enter the following commands.

```
CE1(config)#interface ethernet 1/1
CE1(config-if-e10000-1/1)# ip rip v2-only
CE1(config-if-e10000-1/1)# ip address 33.33.33.2
```

Configuring the VRF on the PE router to redistribute RIP routes

To allow RIP route exchange between a specified VRF on a PE router and its associated CE router, the VRF must be enabled redistribute RIP routes. To enable the VRF VPN1 on PE 1 router in Figure 209 to redistribute RIP routes, enter the following commands.

```
PE1(config-bgp)# address-family ipv4 unicast vrf VPN1
PE1(config-bgp-ipv4u-vrf)# redistribute rip
```

Configuring RIP on the PE router to redistribute BGP-VPNv4 routes

To allow RIP route exchange between a specified VRF on a PE router and its associated CE router, RIP must be configured to redistribute BGP routes from the local AS. To enable RIP on PE 1 in Figure 209 and configure it to redistribute BGP-VPNv4 routes into RIP, enter the following commands.

```
PE1(config)# router rip vrf VPN1
PE1(config-rip-router)# redistribute bgp
```

Enabling RIP on the PE router interface

To allow RIP route exchange between a PE router and its associated CE router, RIP must be enabled on the PE's interface that is associated with the VRF and connected to the PE router. To configure RIP on the interface of the PE 1 router that is associated with VRF VPN1 in Figure 209 to CE 1, enter the following commands.

```
PE1(config)#interface ethernet 1/1
PE1(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE1(config-if-e10000-1/1)# ip address 33.33.33.3/24
PE1(config-if-e10000-1/1)# ip rip v2-only
```

RIP to CE example

In this example, the network shown in Figure 209 is configured for RIP to forward routes between the networks attached to the CE routers and the PE routers. IBGP is used to forward routes between the PE routers and an LSP tunnel is configured across the MPLS Domain. Figure 209 contains all of the network addresses and AS numbers required to perform this configuration. The configurations are shown for only the CE 1, CE 5, PE 1 and PE 5 routers, which demonstrates what is required to make VPN1 work. The configurations for VPN2, VPN3, and VPN 4 would essentially be the same.

CE 1 configuration

This configuration example describes what is required to operate the CE 1 router in Figure 209. In this example, a static route is configured between the external network (10.1.2.0/24) and the loopback interface. RIP is configured to redistribute static routes between the CE 1 router and the attached interface of PE 1.

```
CE1(config)#interface loopback 1
CE1(config-lbif-1)# ip address 33.33.33.1/32
CE1(config-lbif-1)# exit

CE1(config)# ip route 10.1.2.0/24 33.33.33.1
CE1(config)# router rip
CE1(config-lbif-1)# redistribute static
CE1(config-lbif-1)# exit
```

```
CE1(config)# interface ethernet 1/1
CE1(config-if-e10000-1/1)# ip rip v2-only
CE1(config-if-e10000-1/1)# ip address 33.33.33.2
CE1(config-if-e10000-1/1)# exit
```

CE 5 configuration

This configuration example describes what is required to operate the CE 5 router in Figure 209. In this example, a static route is configured between the external network (10.1.3.0/24) and the loopback interface. RIP is configured to redistribute static routes between the CE 5 router and the attached interface of PE 4.

```
CE5(config)#interface loopback 1
CE5(config-lbif-1)# ip address 33.33.36.1/32
CE5(config-lbif-1)# exit

CE5(config)# ip route 10.1.3.0/24 33.33.36.1
CE5(config)# router rip
CE5(config-rip-router)# redistribute static
CE5(config-rip-router)# exit

CE5(config)#interface ethernet 1/1
CE5(config-if-e10000-1/1)# ip rip v2-only
CE5(config-if-e10000-1/1)# ip address 33.33.36.2
CE5(config-if-e10000-1/1)# exit
```

PE 1 configuration

This configuration example describes what is required to operate the PE 1 router in Figure 209. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 1. RIP is configured on the VRF named VPN1 to exchange routes with CE 1 and to redistribute routes from across the MPLS Domain.

IBGP with extended community attributes is configured between PE 1 and PE 4. The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signalled LSP named “tunnel1” to PE 1.

```
PE1(config)#interface loopback 1
PE1(config-lbif-1)# ip address 2.2.2.1/32
PE1(config-lbif-1)# ip ospf area 0
PE1(config-lbif-1)# exit

PE1(config)#ip vrf VPN1
PE1(config-vrf-vpn1)# rd 1:1
PE1(config-vrf-vpn1)# route-target export 100:1
PE1(config-vrf-vpn1)# route-target import 100:2
PE1(config-vrf-vpn1)# exit-vrf

PE1(config)# router bgp
PE1(config-bgp)# local-as 1
PE1(config-bgp)# neighbor 2.2.2.2 remote-as 1
PE1(config-bgp)# neighbor 2.2.2.2 update-source loopback 1
PE1(config-bgp)# address-family vpnv4 unicast
PE1(config-bgp-vpnv4u)# neighbor 2.2.2.2 activate
PE1(config-bgp)# address-family ipv4 unicast vrf VPN1
PE1(config-bgp-ipv4u-vrf)# redistribute rip
PE1(config-bgp-ipv4u-vrf)# exit
```

```
PE1(config)# router rip vrf VPN1
PE1(config-rip-router)# redistribute bgp
PE1(config-rip-router)# exit

PE1(config)# router ospf
PE1(config-ospf-router)# area 0
PE1(config-ospf-router)#

PE1(config)#interface ethernet 1/2
PE1(config-if-e10000-1/2)# ip address 33.33.34.1/24
PE1(config-if-e10000-1/2)# ip ospf area 0
PE1(config-if-e10000-1/2)# exit

PE1(config)# router mpls
PE1(config-mpls)# policy
PE1(config-mpls-policy)# traffic-engineering ospf
PE1(config-mpls)# mpls-interface eth 1/2
PE1(config-mpls)# lsp tunnel1
PE1(config-mpls-lsp-tunnel1)# to 2.2.2.2
PE1(config-mpls-lsp-tunnel1)# enable
PE1(config-mpls-lsp-tunnel1)# exit

PE1(config)#interface ethernet 1/1
PE1(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE1(config-if-e10000-1/1)# ip address 33.33.33.3/24
PE1(config-if-e10000-1/1)# ip rip v2-only
PE1(config-if-e10000-1/1)# exit
```

PE 4 configuration

This configuration example describes what is required to operate the PE 4 router in Figure 209. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 5. RIP is configured on the VRF named VPN1 to exchange routes with CE 1 and to redistribute routes from across the MPLS Domain.

IBGP with extended community attributes is configured between PE 4 and PE 1. The OSPF area is specified as 0, and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signaled LSP named “tunnel1” to PE 1.

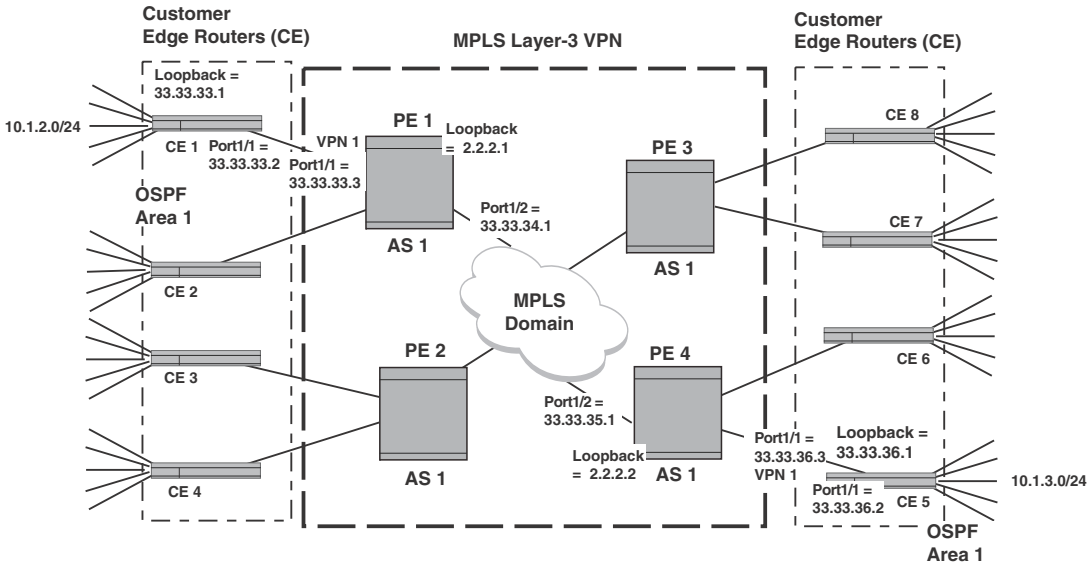
```
PE4(config)# interface loopback 1
PE4(config-lbif-1)# ip address 2.2.2.2/24
PE4(config-lbif-1)# ip ospf area 0
PE4(config-lbif-1)# exit

PE4(config)#ip vrf VPN1
PE4(config-vrf-vpn1)# rd 1:2
PE4(config-vrf-vpn1)# route-target export 100:1
PE4(config-vrf-vpn1)# route-target import 100:2
PE4(config-vrf-vpn1)# exit-vrf

PE4(config)# router bgp
PE4(config-bgp)# local-as 1
PE4(config-bgp)# neighbor 2.2.2.1 remote-as 1
PE4(config-bgp)# neighbor 2.2.2.1 update-source loopback 1
PE4(config-bgp)# address-family vpnv4 unicast
PE4(config-bgp-vpnv4u)# neighbor 2.2.2.1 activate
PE4(config-bgp)# address-family ipv4 unicast vrf VPN1
PE4(config-bgp-ipv4u-vrf)# redistribute rip
PE4(config-bgp-ipv4u-vrf)# exit
```

```
PE4(config)# router rip vrf VPN1
PE4(config-rip-router)# redistribute bgp
PE4(config-rip-router)# exit

PE4(config)# router ospf
PE4(config-ospf-router)# area 0
PE4(config-ospf-router)# exit

PE4(config)#interface ethernet 1/2
PE4(config-if-e10000-1/2)# ip address 33.33.35.1/24
PE4(config-if-e10000-1/2)# ip ospf area 0
PE4(config-if-e10000-1/2)# exit

PE4(config)# router mpls
PE4(config-mpls)# policy
PE4(config-mpls-policy)# traffic-engineering ospf
PE4(config-mpls)# mpls-interface eth 1/2
PE4(config-mpls)# lsp tunnel1
PE4(config-mpls-lsp-tunnel1)# to 2.2.2.1
PE4(config-mpls-lsp-tunnel1)# enable
PE4(config-mpls-lsp-tunnel1)# exit

PE4(config)#interface ethernet 1/1
PE4(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE4(config-if-e10000-1/1)# ip address 33.33.36.3/24
PE4(config-if-e10000-1/1)# ip rip v2-only
PE4(config-if-e10000-1/1)# exit
```

OSPF for route exchange

OSPF can be used to exchange routes between CE routers and PE routers. In this situation, OSPF must be enabled on the CE router with a local area and enabled on the interface that is connected to the interface of the PE that is associated with the VRF that you want to advertise OSPF routes on. On the PE router, the VRF must be enabled in BGP to redistribute OSPF routes, and OSPF must be enabled for the VRF and configured to redistribute routes from BGP-VPNv4. Figure 210 provides an example of a network where OSPF is used to exchange routes between CE routers and PE routers.

FIGURE 210 OSPF to CE network example



To configure OSPF to exchange routes between PE routers and CE routers, you must perform the configuration steps listed below.

1. "Configuring OSPF on the CE router".
2. "Enabling OSPF on the CE router interface".
3. "Configuring the VRF on the PE router to redistribute OSPF routes".
4. "Configuring OSPF on the PE router to redistribute BGP-VPNv4 routes".
5. "Enabling OSPF on the PE router interface".

Configuring OSPF on the CE router

To allow OSPF route exchange between a CE router and its associated PE router, OSPF must be enabled on the CE router. To configure OSPF on the CE 1 router for local area 1 in Figure 210 and enable it to redistribute static routes through OSPF, enter the following commands.

```
CE1(config)# router ospf
CE1(config-ospf-router)# area 1
CE1(config-ospf-router)# redistribute static
```

Enabling OSPF on the CE router interface

To allow OSPF route exchange between a CE router and its associated PE router, OSPF must be enabled on the interface that connects to the VRF-enabled interface of its associated PE router. To configure OSPF on the interface of the CE 1 router in Figure 210 that is connected to the VRF VPN1 associated interface on PE 1, enter the following commands.

```
CE1(config)#interface ethernet 1/1
CE1(config-if-e10000-1/1)# ip ospf area 1
CE1(config-if-e10000-1/1)# ip address 33.33.33.2
```

Configuring the VRF on the PE router to redistribute OSPF routes

To allow OSPF route exchange between a specified VRF on a PE router and its associated CE router, the VRF must be enabled to redistribute OSPF routes. To enable the VRF VPN1 on PE 1 router in Figure 210 to redistribute OSPF routes, enter the following commands.

```
PE1(config-bgp)# address-family ipv4 unicast vrf VPN1
PE1(config-bgp-ipv4u-vrf)# redistribute ospf match internal
PE1(config-bgp-ipv4u-vrf)# redistribute ospf match external1
PE1(config-bgp-ipv4u-vrf)# redistribute ospf match external2
```

Configuring OSPF on the PE router to redistribute BGP-VPNv4 routes

To allow OSPF route exchange between a specified VRF on a PE router and its associated CE router, OSPF must be configured to redistribute BGP routes from the local AS. To enable OSPF on PE 1 in Figure 210 and configure it to redistribute BGP-VPNv4 routes into OSPF, enter the following commands.

```
PE1(config)# router ospf vrf VPN1
PE1(config-ospf-router)# domain-id 0.0.0.100
PE1(config-ospf-router)# domain-tag 1200
PE1(config-ospf-router)# area 1
PE1(config-ospf-router)# redistribute bgp
```

Enabling OSPF on the PE router interface

To allow OSPF route exchange between a PE router and its associated CE router, OSPF must be enabled on the PE's interface that is associated with the VRF and connected to the PE router. To configure OSPF on the interface of the PE 1 router that is associated with VRF VPN1 in Figure 210 to CE 1, enter the following commands.

```
PE1(config)#interface ethernet 1/1
PE1(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE1(config-if-e10000-1/1)# ip address 33.33.33.3/24
PE1(config-if-e10000-1/1)# ip ospf area 1
```

OSPF to CE example

In this example, the network shown in Figure 210 is configured for OSPF to forward routes between the networks attached to the CE routers and the PE routers. IBGP is used to forward routes between the PE routers and an LSP tunnel is configured across the MPLS Domain. Figure 210 contains all of the network addresses and AS numbers required to perform this configuration. The configurations are shown for only the CE 1, CE 5, PE 1 and PE 5 routers, which demonstrates what is required to make VPN1 work. The configurations for VPN2, VPN3, and VPN 4 would essentially be the same.

CE 1 configuration

This configuration example describes what is required to operate the CE 1 router in Figure 210. In this example, a static route is configured between the external network (10.1.2.0/24) and the loopback interface. OSPF is configured to redistribute static routes between the CE 1 router and the attached interface of PE 1.

```
CE1(config)#interface loopback 1
CE1(config-lbif-1)# ip address 33.33.33.1/32
CE1(config-lbif-1)# exit

CE1(config)# ip route 10.1.2.0/24 33.33.33.1
CE1(config)# router ospf
CE1(config-ospf-router)# area 1
CE1(config-ospf-router)# redistribute static
CE1(config-ospf-router)# exit

CE1(config)#interface ethernet 1/1
CE1(config-if-e10000-1/1)# ip address 33.33.33.2
CE1(config-if-e10000-1/1)# ip ospf area 1
CE1(config-if-e10000-1/1)# exit
```

CE 5 configuration

This configuration example describes what is required to operate the CE 5 router in Figure 210. In this example, a static route is configured between the external network (10.1.3.0/24) and the loopback interface. OSPF is configured to redistribute static routes between the CE 5 router and the attached interface of PE 4.

```
CE5(config)#interface loopback 1
CE5(config-lbif-1)# ip address 33.33.36.1/32
CE5(config-lbif-1)# exit

CE5(config)# ip route 10.1.3.0/24 33.33.36.1
CE5(config)# router ospf
CE5(config-ospf-router)# area 1
CE5(config-ospf-router)# redistribution static
CE5(config-ospf-router)# exit

CE5(config)#interface ethernet 1/1
CE5(config-if-e10000-1/1)# ip address 33.33.36.2
CE5(config-if-e10000-1/1)# ip ospf area 1
CE5(config-if-e10000-1/1)# exit
```

PE 1 configuration

This configuration example describes what is required to operate the PE 1 router in Figure 210. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 1. OSPF is configured on the VRF named VPN1 to exchange routes with CE 1 and to redistribute routes from across the MPLS Domain.

IBGP with extended community attributes is configured between PE 1 and PE 4. The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signalled LSP named “tunnel1” to PE 1.

```
PE1(config)#interface loopback 1
PE1(config-lbif-1)# ip address 2.2.2.1/24
PE1(config-lbif-1)# ip ospf area 0
PE1(config-lbif-1)# exit

PE1(config)#ip vrf VPN1
PE1(config-vrf-vpn1)# rd 1:1
PE1(config-vrf-vpn1)# route-target export 100:1
PE1(config-vrf-vpn1)# route-target import 100:2
```

```
PE1(config-vrf-vpn1)# exit-vrf

PE1(config)# router bgp
PE1(config-bgp)# local-as 1
PE1(config-bgp)# neighbor 2.2.2.2 remote-as 1
PE1(config-bgp)# neighbor 2.2.2.2 update-source loopback 1
PE1(config-bgp)# address-family vpnv4 unicast
PE1(config-bgp-vpnv4u)# neighbor 2.2.2.2 activate
PE1(config-bgp)# address-family ipv4 unicast vrf VPN1
PE1(config-bgp-ipv4u-vrf)# redistribute ospf
PE1(config-bgp-ipv4u-vrf)# exit

PE1(config)# router ospf vrf VPN1
PE1(config-ospf-router)# domain-id 0.0.0.100
PE1(config-ospf-router)# domain-tag 0.0.0.100
PE1(config-ospf-router)# area 1
PE1(config-ospf-router)# redistribute bgp
PE1(config-ospf-router)# exit

PE1(config)# router ospf
PE1(config-ospf-router)# area 0
PE1(config-ospf-router)# exit

PE1(config)#interface ethernet 1/2
PE1(config-if-e10000-1/2)# ip address 33.33.34.1/24
PE1(config-if-e10000-1/2)# ip ospf area 0
PE1(config-if-e10000-1/2)# exit

PE1(config)# router mpls
PE1(config-mpls)# policy
PE1(config-mpls-policy)# traffic-engineering ospf
PE1(config-mpls)# mpls-interface eth 1/2
PE1(config-mpls)# lsp tunnel1
PE1(config-mpls-lsp-tunnel1)# to 2.2.2.2
PE1(config-mpls-lsp-tunnel1)# enable
PE1(config-if-e10000-1/2)# exit

PE1(config)#interface ethernet 1/1
PE1(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE1(config-if-e10000-1/1)# ip address 33.33.33.3/24
PE1(config-if-e10000-1/1)# ip ospf area 1
PE1(config-if-e10000-1/1)# exit
```

PE 4 configuration

This configuration example describes what is required to operate the PE 4 router in Figure 210. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 5. OSPF is configured on the VRF named VPN1 to exchange routes with CE 5 and to redistribute routes from across the MPLS Domain.

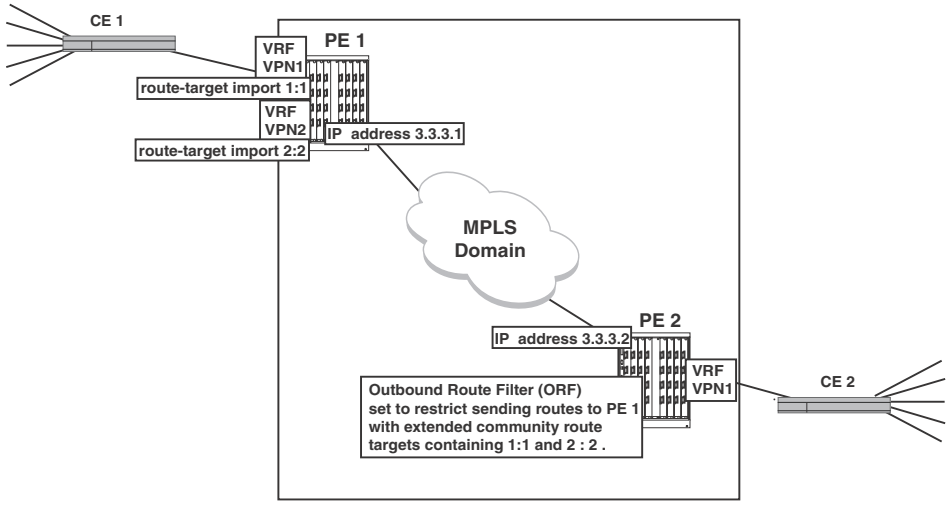
```
PE4(config)#interface loopback 1
PE4(config-lbif-1)# ip address 2.2.2.2/32
PE1(config-lbif-1)# ip ospf area 1
PE1(config-lbif-1)# exit

PE4(config)#ip vrf VPN1
PE4(config-vrf-vpn1)# rd 2:1
```

```
PE4(config-vrf-vpn1)# route-target export 100:1
PE4(config-vrf-vpn1)# route-target import 100:2
PE4(config-vrf-vpn1)# exit-vrf

PE4(config)# router bgp
PE4(config-bgp)# local-as 1
PE4(config-bgp)# neighbor 2.2.2.1 remote-as 1
PE4(config-bgp)# neighbor 2.2.2.1 update-source loopback 1
PE4(config-bgp)# address-family vpnv4 unicast
PE4(config-bgp-vpnv4u)# neighbor 2.2.2.1 activate
PE4(config-bgp)# address-family ipv4 unicast vrf VPN1
PE4(config-bgp-ipv4u-vrf)# redistribute ospf
PE4(config-bgp-ipv4u-vrf)# exit

PE4(config)# router ospf vrf VPN1
PE4(config-ospf-router)# domain-id 0.0.0.100
PE4(config-ospf-router)# domain-tag 0.0.0.100
PE4(config-ospf-router)# area 1
PE4(config-ospf-router)# redistribute bgp
PE4(config-ospf-router)# exit

PE4(config)# router ospf
PE4(config-ospf-router)# area 0
PE4(config-ospf-router)# exit

PE4(config)#interface ethernet 1/2
PE4(config-if-e10000-1/2)# ip ospf area 0
PE4(config-if-e10000-1/2)# ip address 33.33.35.1/24
PE4(config-if-e10000-1/2)#exit

PE4(config)# router mpls
PE4(config-mpls)# policy
PE4(config-mpls-policy)# traffic-engineering ospf
PE4(config-mpls)# mpls-interface eth 1/2
PE4(config-mpls)# lsp tunnel1
PE4(config-mpls-lsp-tunnel1)# to 2.2.2.1
PE4(config-mpls-lsp-tunnel1)# enable
PE4(config-mpls-lsp-tunnel1)# exit

PE4(config)# interface ethernet 1/1
PE4(config-if-e10000-1/1)# ip vrf forwarding VPN1
PE4(config-if-e10000-1/1)# ip address 33.33.36.3/24
PE4(config-if-e10000-1/1)# ip ospf area 1
PE4(config-if-e10000-1/1)# exit
```

Cooperative route filtering

The ***cooperative route filtering*** feature allows you to move the filtering function of a route-target import filter to a peer. In this situation, an Outbound Route Filter (ORF) is derived from the contents of all of the ***route-target import*** commands of BGP configured VRFs on a PE and shared with a peer PE. This ORF is then used to exclude any routes that are blocked by that ORF from being sent by the peer PE to the PE with the ***route-target import*** commands that the ORF was derived from. For example, in Figure 211 the routes that are admitted into VPN1 and VPN2 have route targets of 1:1 and 2:2. You can use the cooperative route filtering feature to send an ORF that is derived from the route-target import commands on PE 1 to PE 2 to only accept these routes.

FIGURE 211 Cooperative route filtering example



The following example shows the commands required to configure VRF VPN1 on PE 1 in Figure 211 with an import route-target of 1:1 and VRF VPN2 on PE 1 with an import route-target import of 2:2.

```
PE1(config)#ip vrf VPN1
PE1(config-ip-vrf-VPN1)# route-target import 1:1
PE1(config-ip-vrf-VPN1)#exit-vrf

PE1(config)#ip vrf VPN2
PE1(config-ip-vrf-VPN2)# route-target import 2:2
PE1(config-ip-vrf-VPN2)#exit-vrf
```

The following commands configure PE 1 to send the filter derived from the import route-target commands in VPN1 and VPN2 to PE 2.

```
PE1(config-bgp)# address-family vpnv4 unicast
PE1(config-bgp-vpnv4u)# neighbor 3.3.3.2 capability orf extended-community
send-vrf-filter
```

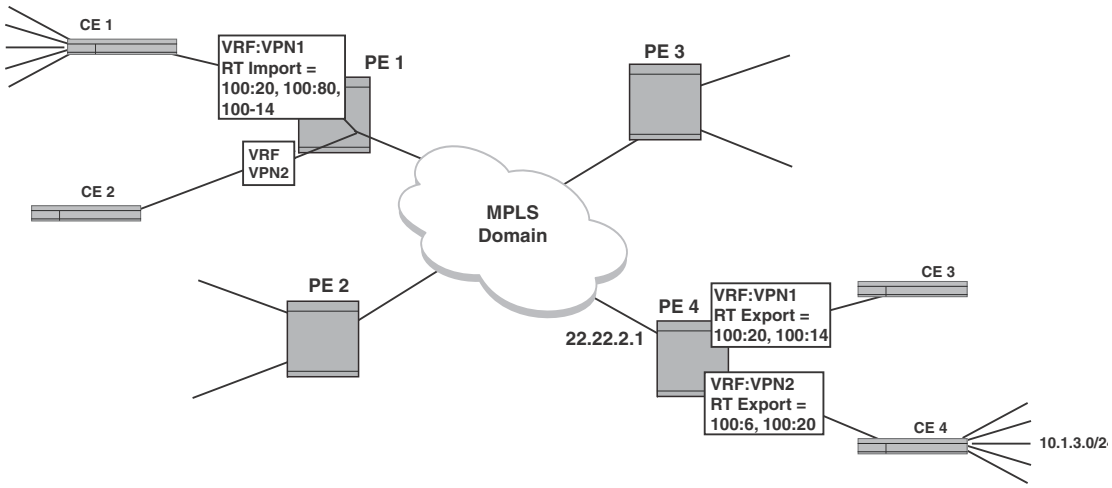
The following commands configure PE 2 to receive the filter derived from the import route-target commands in VPN1 and VPN2 on PE 1.

```
PE2(config-bgp)# address-family vpnv4 unicast
PE2(config-bgp-ivpnv4u)# neighbor 3.3.3.1 capability orf extended-community
receive
```

Using an IP extcommunity variable with route map

In Figure 212, the VRF named “VPN1” on PE 1 is set to import routes with RT 100:14, 100:20 and 100:80. The VRF named “VPN1” on PE 4 is configured to export routes with RT 100:20 and 100:14. The VRF named “VPN2” on PE 4 is configured to export routes with RT 100:6 and 100:20. A route-map is configured from a BGP neighbor command on PE 1 to not install all routes from PE 4 with RT 100:6. This will block all routes from VPN2 being sent to PE 1.

FIGURE 212 IP Extcommunity and route-map usage



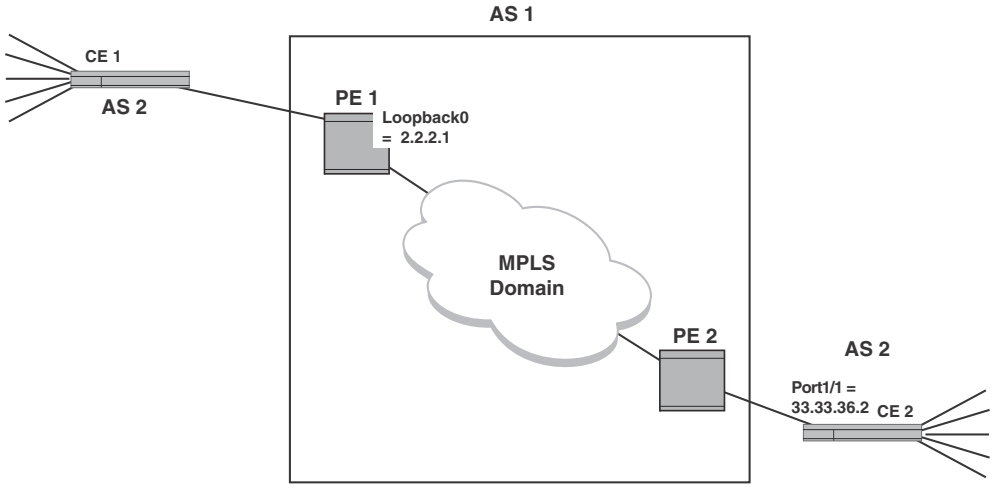
The following example shows the configuration commands required on the PE 1 router for the example shown in Figure 212. In this example, the route-map ExcludeRoute has a **match extcommunity** value that references the extcommunity 20. The **extended community list 20** command specifies that routes with RT 100:6 are to be denied. The **neighbor route-map** command exports the ExcludeRoute route-map to the BGP neighbor PE 4. Consequently, PE 4 will block the export or route-target 100:6 to PE 1. This will block all routes from VPN2 on PE 4 from being sent to PE 1.

```
PE1(config)# router bgp
PE1(config-bgp)# local-as 100
PE1(config-bgp)# neighbor 22.22.2.1 remote-as 100
PE1(config-bgp)# address-family vpnv4 unicast
PE1(config-bgp-vpnv4u)# neighbor 22.22.2.1 activate
PE1(config-bgp-vpnv4u)# neighbor 22.22.2.1 route-map in ExcludeRoute
PE1(config-bgp-vpnv4u)# neighbor 22.22.2.1 send-community extended
PE1(config-bgp-vpnv4u)# exit

PE1(config)# route-map ExcludeRoute permit 10
PE1(config-routemap ExcludeRoute)# match extcommunity 20
PE1(config-routemap ExcludeRoute)# exit

PE1(config)#ip extcommunity-list 20 deny RT 100:6
PE1(config)#ip vrf VPN1
PE1(config-ip-vrf-vpn1)#rd 1:1
PE1(config-ip-vrf-vpn1)#route-target import 100:20
PE1(config-ip-vrf-vpn1)#route-target import 100:80
PE1(config-ip-vrf-vpn1)#route-target import 100:14
PE1(config-ip-vrf-vpn1)#exit-vrf
```

Autonomous system number override

In the example shown in Figure 213 the service providers network is in AS1 and the customer wants both of his CE routers at different sites to use AS 2. When a route is sent from CE 1 to CE 2, it will contain an AS_PATH attribute containing AS 2. When CE 2 sees that the AS_PATH attribute contains its own AS number, it will reject the route.

FIGURE 213 AS number override example



One solution to this problem is to configure PE 2 to override the AS_PATH attribute that contains AS 2. When this is enabled, the PE router determines when the AS_PATH attribute in a route intended for a neighbor CE contains the same AS number as the CE. When this is determined, the PE router substitutes its own AS number for the CE's in the AS_PATH attribute. The CE is then able to receive the route. The following addition conditions apply when this feature is in effect:

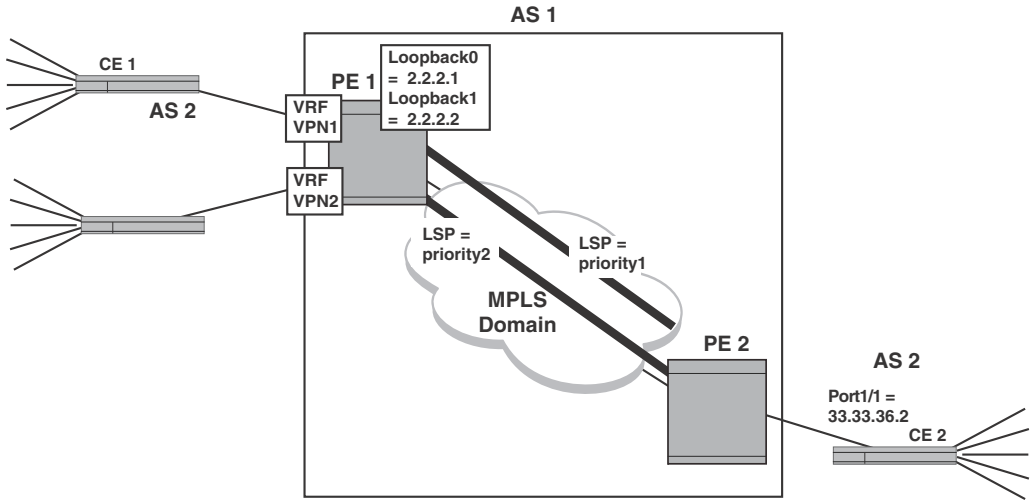
The following example describes the configuration of PE 2 required to enable Autonomous System number override for the BGP neighbor CE 2.

```
PE2(config)# router bgp
PE2(config-bgp)# local-as 1
PE2(config-bgp)# neighbor 2.2.2.1 remote-as 1
PE2(config-bgp)# neighbor 2.2.2.1 update-source loopback0
PE2(config-bgp)# address-family ipv4 unicast vrf VPN1
PE2(config-bgp-ipv4u-vrf)# neighbor 33.33.36.2 remote-as 2
PE2(config-bgp-ipv4u-vrf)# neighbor 33.33.36.2 as-override
```

Setting an LSP for each VRF on a PE

Figure 214 provides an example of assigning a different LSP for each VRF on a PE. In this example, PE 1 contains two VRFs: VPN1 and VPN2. It also contains two loopback interfaces with the following IP addresses: Loopback 0 = 2.2.2.1 and Loopback 2 = 2.2.2.2. Nexthop addresses for VPN1 and VPN2 can be created separately to Loopback 0 and Loopback 1. Then, different LSPs are assigned to each of the Loopback addresses.

FIGURE 214 Support per-VRF BGP nexthop



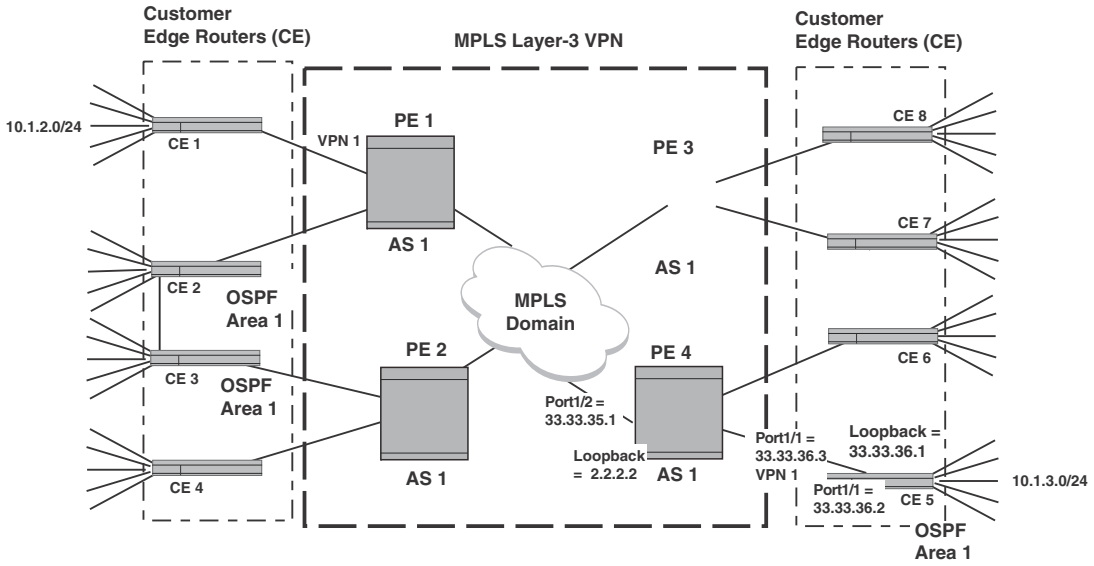
The following configuration example shows the elements in the PE 2 configuration required to make this example operate.

```
PE2(config)# ip vrf VPN1
PE2(config-ip-vrf-vpn1)# bgp next-hop loopback 0
PE2(config-ip-vrf-vpn1)#exit-vrf
PE2(config)# ip vrf VPN2
PE2(config-ip-vrf-vpn2)# bgp next-hop loopback 1
PE2(config-ip-vrf-vpn2)# exit-vrf

PE2(config)# router mpls
PE2(config-mpls)# mpls-interface ethe 1/1
PE2(config-mpls)# lsp priority1
PE2(config-mpls-lsp-priority1)# to 2.2.2.2
PE2(config-mpls-lsp-priority1)# primary-path prim-path1
PE2(config-mpls-lsp-priority1)# secondary-path sec-path1
PE2(config-mpls-lsp-priority1)# enable
PE2(config-mpls)# lsp priority2
PE2(config-mpls-lsp-priority2)# to 2.2.2.1
PE2(config-mpls-lsp-priority2)# primary prim-path2
PE2(config-mpls-lsp-priority2)# secondary sec-path2
PE2(config-mpls-lsp-priority2)# enable
```

OSPF sham links

In the example shown in Figure 215, CE 2 and CE 3 are both in OSPF Area 1 and connect to the same service provider network through different PEs. An additional backdoor connection is configured between them over another network. OSPF recognizes the backdoor connection as an Intra-area connection and the connection through the service provider network as an Inter-network connection. Because OSPF favors Intra-area routes over Inter-network routes, most traffic between CE 2 and CE 3 travels across the backdoor link. If this is the preferred link in the network, the configuration is as it should be. However, if you prefer traffic between the two networks to be routed across the service provider network, this configuration can causes problems.

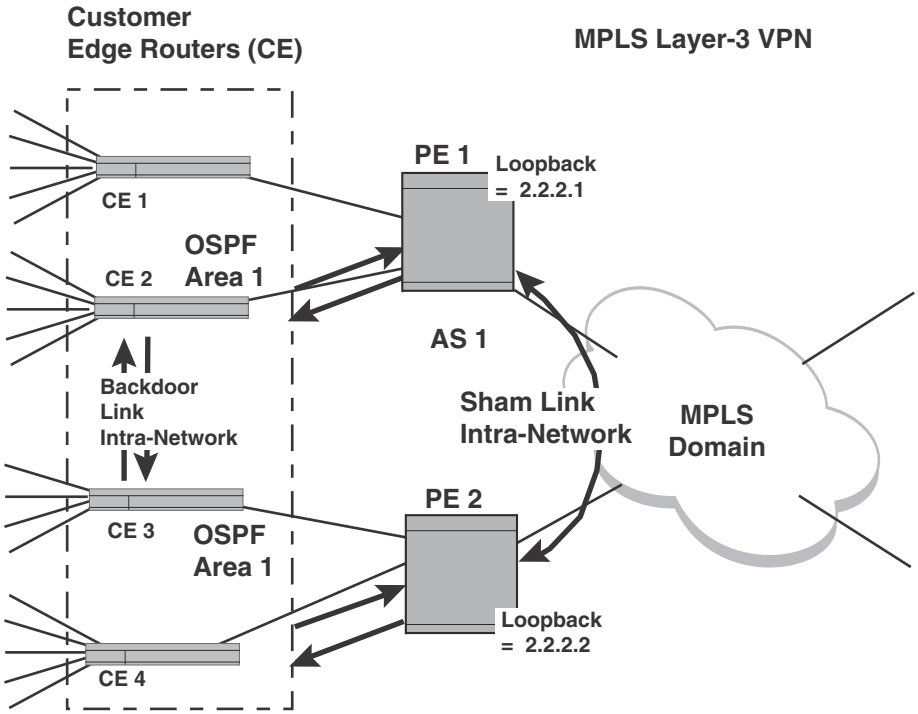
FIGURE 215 BGP or MPLS VPN with OSPF backdoor link



Problems can be avoided by creating a virtual intra-area OSPF link between two PEs. This virtual link is called a sham link. A sham link directs OSPF to treat the route through the service provider network as an intra-area link. A cost is assigned to the sham link to help the OSPF network determine when to route over the sham link route and when to use the backdoor link. Because this virtual link (sham-link) is an intra-area link, the OSPF areas in which each of the PEs reside must be the same.

NOTE
For sham links to work, OSPF cannot be configured on the loopback interface in the applicable area.

FIGURE 216 BGP or MPLS VPN with OSPF including Sham link and backdoor link



This configuration example describes the additional configuration required to create a sham link between PE 1 and PE 2 in the example shown in Figure 216. In this example, the VRF VPN1 is added to the loopback interface configuration, and a sham link with a cost of 10 is created between the loopback interfaces on PE 1 and PE 2.

After this configuration is implemented, routes between CE 2 and CE 3 over the service provider network will be preferred to the backdoor link that exists between these CEs.

PE 1 configuration

```
PE1(config)#interface loopback 1
PE1(config-lbif-1)#ip vrf forwarding VPN1
PE1(config-lbif-1)# ip address 2.2.2.1/24
PE1(config)#ip vrf VPN1
PE1(config)# router ospf vrf VLAN1
PE1(config-ospf-router)# area 1 sham-link 2.2.2.1 2.2.2.2 cost 10
PE1(config-ospf-router)# redistribution bgp
```

PE 2 configuration

```
PE2(config)#interface loopback 1
PE2(config-lbif-1)#ip vrf forwarding VPN1
PE2(config-lbif-1)# ip address 2.2.2.2/24
PE2(config)#ip vrf VPN1
PE2(config)# router ospf vrf VLAN1
PE2(config-ospf-router)# area 1 sham-link 2.2.2.2 2.2.2.1 cost 10
PE2(config-ospf-router)# redistribution bgp
```

IPv6 Addressing 37

IPv6 addressing overview

PowerConnect B-MLXe supports the following IPv6 Addressing features:

- IPv6 Addressing
- IPv6 Address Unicast
- IPv6 Address Multicast
- IPv6 Address Anycast
- IPv6 Stateless auto-configuration
- IPv6 Address in the Configuration

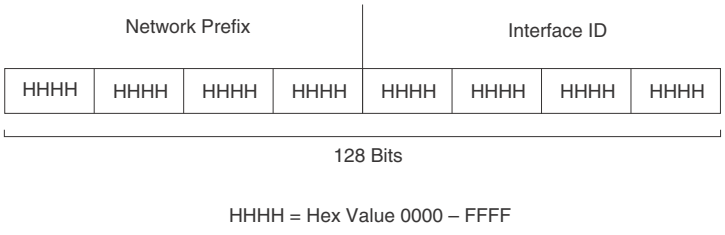
This chapter includes overview information about the following topics:

- IPv6 addressing.
- The IPv6 stateless auto-configuration feature, which enables a host on a local link to automatically configure its interfaces with new and globally unique IPv6 addresses associated with its location.

A limitation of IPv4 is its 32-bit addressing format, which is unable to satisfy potential increases in the number of users, geographical needs, and emerging applications. To address this limitation, IPv6 introduces a new 128-bit addressing format.

An IPv6 address is composed of 8 fields of 16-bit hexadecimal values separated by colons (:). Figure 217 shows the IPv6 address format.

FIGURE 217 IPv6 address format



As shown in Figure 217, HHHH is a 16-bit hexadecimal value, while H is a 4-bit hexadecimal value. The following is an example of an IPv6 address:

2001:0000:0000:0200:002D:D0FF:FE48:4672

Note that the sample IPv6 address includes hexadecimal fields of zeros. To make the address less cumbersome, you can do the following:

- Omit the leading zeros; for example, 2001:0:0:200:2D:D0FF:FE48:4672.

- Compress the successive groups of zeros at the beginning, middle, or end of an IPv6 address to two colons (::) once per address; for example, 2001::200:2D:D0FF:FE48:4672.

When specifying an IPv6 address in a command syntax, keep the following in mind:

- You can use the two colons (::) once in the address to represent the longest successive hexadecimal fields of zeros.
- The hexadecimal letters in the IPv6 addresses are not case-sensitive.

As shown in Figure 217, the IPv6 network prefix is composed of the left-most bits of the address. As with an IPv4 address, you can specify the IPv6 prefix using the <*prefix*> or <*prefix-length*> format, where the following applies:

The <*prefix*> parameter is specified as 16-bit hexadecimal values separated by a colon.

The <*prefix-length*> parameter is specified as a decimal value that indicates the left-most bits of the IPv6 address.

The following is an example of an IPv6 prefix:

2001:FF08:49EA:D088::/64

IPv6 address types

As with IPv4 addresses, you can assign multiple IPv6 addresses to a router interface. Table 300 presents the three major types of IPv6 addresses that you can assign to a router interface.

A major difference between IPv4 and IPv6 addresses is that IPv6 addresses support ***scope***, which describes the topology in which the address may be used as a unique identifier for an interface or set of interfaces.

Unicast and multicast addresses support scoping as follows:

- **Unicast addresses support two types of scope:** global scope and local scope. In turn, local scope supports site-local addresses and link-local addresses. Table 300 describes global, site-local, and link-local addresses and the topologies in which they are used.
- Multicast addresses support a scope field, which Table 300 describes.

TABLE 300 IPv6 address types

Address type	Description	Address structure
Unicast	An address for a single interface. A packet sent to a unicast address is delivered to the interface identified by the address.	Depends on the type of the unicast address: • Aggregatable global address — An address equivalent to a global or public IPv4 address. The address structure is as follows: a fixed prefix of 2000::/3 (001), a 45-bit global routing prefix, a 16-bit subnet ID, and a 64-bit interface ID. • Site-local address — An address used within a site or intranet. (This address is similar to a private IPv4 address.) A site consists of multiple network links. The address structure is as follows: a fixed prefix of FEC0::/10 (1111 1110 11), a 16-bit subnet ID, and a 64-bit interface ID. • Link-local address — An address used between directly connected nodes on a single network link. The address structure is as follows: a fixed prefix of FE80::/10 (1111 1110 10) and a 64-bit interface ID. • IPv4-compatible address — An address used in IPv6 transition mechanisms that tunnel IPv6 packets dynamically over IPv4 infrastructures. The address embeds an IPv4 address in the low-order 32 bits and the high-order 96 bits are zeros. The address structure is as follows: 0:0:0:0:0:0:A.B.C.D. • Loopback address — An address (0:0:0:0:0:0:0:1 or ::1) that a router can use to send an IPv6 packet to itself. You cannot assign a loopback address to a physical interface. • Unspecified address — An address (0:0:0:0:0:0:0:0 or ::) that a node can use until you configure an IPv6 address for it.
Multicast	An address for a set of interfaces belonging to different nodes. Sending a packet to a multicast address results in the delivery of the packet to all interfaces in the set.	A multicast address has a fixed prefix of FF00::/8 (1111 1111). The next 4 bits define the address as a permanent or temporary address. The next 4 bits define the scope of the address (node, link, site, organization, global).
Anycast	An address for a set of interfaces belonging to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface identified by the address.	An anycast address looks similar to a unicast address, because it is allocated from the unicast address space. If you assign a unicast address to multiple interfaces, it is an anycast address. An interface assigned an anycast address must be configured to recognize the address as an anycast address. An anycast address can be assigned to a router only. An anycast address must not be used as the source address of an IPv6 packet.

A router automatically configures a link-local unicast address for an interface by using the prefix of FE80::/10 (1111 1110 10) and a 64-bit interface ID. The 128-bit IPv6 address is then subjected to duplicate address detection to ensure that the address is unique on the link. If desired, you can override this automatically configured address by explicitly configuring an address. For more information about explicitly configuring this address, refer to “Configuring IPv6 on each interface” on page 1718.

IPv6 stateless auto-configuration

Dell devices use the IPv6 stateless auto-configuration feature to enable a host on a local link to automatically configure its interfaces with new and globally unique IPv6 addresses associated with its location. The automatic configuration of a host interface is performed without the use of a server, such as a Dynamic Host Configuration Protocol (DHCP) server, or manual configuration.

The automatic configuration of a host interface works in the following way: a router on a local link periodically sends router advertisement messages containing network-type information, such as the 64-bit prefix of the local link and the default route, to all nodes on the link. When a host on the link receives the message, it takes the local link prefix from the message and appends a 64-bit interface ID, thereby automatically configuring its interface. (The 64-bit interface ID is derived from the MAC address of the host's NIC.) The 128-bit IPv6 address is then subjected to duplicate address detection to ensure that the address is unique on the link.

The duplicate address detection feature verifies that a unicast IPv6 address is unique before it is assigned to a host interface by the stateless auto configuration feature. Duplicate address detection uses neighbor solicitation messages to verify that a unicast IPv6 address is unique. For more information about duplicate address detection, refer to "Setting neighbor solicitation parameters for duplicate address detection" on page 1735.

NOTE
For the stateless auto configuration feature to work properly, the advertised prefix length in router advertisement messages must always be 64 bits. For more information about the router advertisement message, refer to "Router advertisement and solicitation messages" on page 1734.

The IPv6 stateless auto-configuration feature can also automatically reconfigure a host's interfaces if you change the ISP for the host's network. (The host's interfaces must be renumbered with the IPv6 prefix of the new ISP.)

The renumbering occurs in the following way: a router on a local link periodically sends advertisements updated with the prefix of the new ISP to all nodes on the link. (The advertisements still contain the prefix of the old ISP.) A host can use the addresses created from the new prefix and the existing addresses created from the old prefix on the link. When you are ready for the host to use the new addresses only, you can configure the lifetime parameters appropriately using the **ipv6 nd prefix-advertisement** command. During this transition, the old prefix is removed from the router advertisements. At this point, only addresses that contain the new prefix are used on the link. For more information about configuring the lifetime parameters, refer to "Controlling prefixes advertised in IPv6 router advertisement messages" on page 1737.

Configuring Basic IPv6 Connectivity 38

PowerConnect B-MLXe supports the following Basic IPv6 Connectivity features.

- IPv6 Routing
- IPv6 Anycast Addresses
- IPv6 Host Support
- Restricting SNMP Access to an IPv6 Node
- IPv6 Domain Name Server (DNS) Resolver
- ECMP Load Sharing for IPv6
- IPv6 ICMP
- DHCPv6
- IPv6 Neighbor Discovery
- IPv6 Source Routing Security
- IPv6 MTU
- Static Neighbor Entries
- Limiting the Number of Hops an IPv6 Packet Can Traverse
- QoS for IPv6 Traffic
- DNS Queries of IPv4 and IPv6 DNS Servers
- Displaying IPv6 Traffic Statistics
- Displaying the IPv6 Route Table

This chapter explains how to get a PowerConnect router up and running with IPv6. To configure basic IPv6 connectivity, you must do the following:

- Enable IPv6 routing globally.
- Configure an IPv6 address or explicitly enable IPv6 on each router interface over which you plan to forward IPv6 traffic.
- Configure IPv4 and IPv6 protocol stacks. (This step is mandatory only if you want a router interface to send and receive both IPv4 and IPv6 traffic.)

The following configuration tasks are optional:

- Configure IPv6 Domain Name Server (DNS) resolver.
- Configure equal-cost mulitpath (ECMP) routing Load Sharing for IPv6.
- Configure IPv6 Internet Control Message Protocol (ICMP).
- Configure the IPv6 neighbor discovery feature.
- Change the IPv6 maximum transmission unit (MTU).
- Configure static neighbor entries.
- Limit the hop count of an IPv6 packet.

- Configure Quality of Service (QoS) for IPv6 traffic.

Enabling IPv6 routing

By default, IPv6 routing is enabled. If forwarding of IPv6 traffic globally on the device has been disabled, you can enable it by entering the following command.

```
NetIron(config)# ipv6 unicast-routing
```

Syntax: [no] ipv6 unicast-routing

To disable the forwarding of IPv6 traffic globally on the device, enter the **no** form of this command.

NOTE
IPv6 routing is enabled by default and therefore does not appear in the configuration.

Configuring IPv6 on each interface

To forward IPv6 traffic on an interface, the interface must have an IPv6 address, or IPv6 must be explicitly enabled. By default, an IPv6 address is not configured on an interface.

If you choose to configure a global or site-local IPv6 address for an interface, IPv6 is also enabled on the interface. Further, when you configure a global or site-local IPv6 address, you must decide on one of the following in the low-order 64 bits:

- A manually configured interface ID.
- An automatically computed EUI-64 interface ID.

If you prefer to assign a link-local IPv6 address to the interface, you must explicitly enable IPv6, which causes a link-local address to be automatically computed for the interface. If preferred, you can override the automatically configured link-local address with an address that you manually configure.

This section provides the following information:

- Configuring a global or site-local address with a manually configured or automatically computed interface ID for an interface.
- Automatically or manually configuring a link-local address for an interface.
- Configuring IPv6 anycast addresses

Configuring a global or site-local IPv6 address

Configuring a global or site-local IPv6 address on an interface does the following:

- Automatically configures an interface ID (a link-local address), if specified.
- Enables IPv6 on that interface.

Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast address assigned to the interface.

- All-nodes link-local multicast group FF02::1
- All-routers link-local multicast group FF02::2

The neighbor discovery feature sends messages to these multicast groups. For more information, refer to "Configuring IPv6 neighbor discovery" on page 1733.

Configuring a global or site-local IPv6 address with a manually configured interface ID

To configure a global or site-local IPv6 address, including a manually configured interface ID, for an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 address 2001:200:12D:1300:240:D0FF:
FE48:4672/64
```

These commands configure the global prefix 2001:200:12d:1300::/64 and the interface ID::240:D0FF:FE48:4672, and enable IPv6 on Ethernet interface 3/1.

Syntax: ipv6 address <*ipv6-prefix*>/<*prefix-length*>

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

Configuring a global or site-local IPv6 address with an automatically computed EUI-64 interface ID

To configure a global or site-local IPv6 address with an automatically computed EUI-64 interface ID in the low-order 64-bits, enter commands such as the following.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 address 2001:200:12D:1300::/64 eui-64
```

These commands configure the global prefix 2001:200:12d:1300::/64 and an interface ID, and enable IPv6 on Ethernet interface 3/1.

Syntax: [**no**] **ipv6 address** <*ipv6-prefix*>/<*prefix-length*> **eui-64**

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **eui-64** keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits.The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Configuring a link-local IPv6 address

To explicitly enable IPv6 on an interface without configuring a global or site-local address for the interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 enable
```

These commands enable IPv6 on Ethernet interface 3/1 and specify that the interface is assigned an automatically computed link-local address.

Syntax: [no] **ipv6 enable**

NOTE
When configuring VLANs that share a common tagged interface with a Virtual Ethernet (VE) interface, it is recommended that you override the automatically computed link-local address with a manually configured unique address for the interface. If the interface uses the automatically computed address, which in the case of VE interfaces is derived from a global MAC address, all VE interfaces will have the same MAC address.

To override a link-local address that is automatically computed for an interface with a manually configured address, enter commands such as the following.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 address FE80::240:D0FF:FE48:4672 link-local
```

These commands explicitly configure the link-local address FE80::240:D0FF:FE48:4672 for Ethernet interface 3/1.

Syntax: [no] **ipv6 address** <*ipv6-address*> **link-local**

You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **link-local** keyword indicates that the router interface should use the manually configured link-local address instead of the automatically computed link-local address.

Configuring IPv6 anycast addresses

In IPv6, an ***anycast*** address is an address for a set of interfaces that belong to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface that has an anycast address.

An anycast address looks similar to a unicast address, because it is allocated from the unicast address space. If you assign an IPv6 unicast address to multiple interfaces, it is an anycast address. On the device, you configure an interface assigned an anycast address to recognize the address as an anycast address.

For example, the following commands configure an anycast address on interface 2/1.

```
NetIron(config)# int e 2/1
NetIron(config-if-e100-2/1)# ipv6 address 2002::6/64 anycast
```

Syntax: [no] **ipv6 address** <*ipv6-prefix*>|<*prefix-length*> [**anycast**]

IPv6 anycast addresses are described in detail in RFC 1884. See RFC 2461 for a description of how the IPv6 Neighbor Discovery mechanism handles anycast addresses.

Configuring the management port for an IPv6 automatic address configuration

You can configure the management port to automatically obtain an IPv6 address. The process is the same for all ports and is described in detail in the "Configuring a global or site-local IPv6 address with an automatically computed EUI-64 interface ID" on page 1719

IPv6 host support

You can configure the device to be an IPv6 host. An IPv6 host has interfaces with IPv6 addresses, but does not have IPv6 routing enabled.

This section lists supported and unsupported IPv6 host features.

IPv6 host supported features

The following IPv6 host features are supported:

- Automatic address configuration

NOTE
Automatic IPv6 address configuration is supported, however, automatic configuration of an IPv6 *global* address is supported only if there is an IPv6 router present on the network. Manual IPv6 address configuration is not supported.

- HTTP/HTTPS over IPv6
- IPv6 ping
- Telnet using an IPv6 address
- TFTP using an IPv6 address
- Trace route using an IPv6 address
- Name to IPv6 address resolution using IPv6 DNS Server
- IPv6 access lists
- IPv6 debugging
- SSH version 1 over IPv6
- SNMP over IPv6
- Logging (Syslog) over IPv6

IPv6 unsupported features

The following IPv6 features are not supported:

- IPv6 routing
- Tunneling
- MLD version 1 and version 2
- IP security

- IPv6 in boot PROM
- IPv6 address configuration using DHCP
- IPv6 TFTP using IPv6 link local address for a TFTP server
- IPv6 link local address is not supported for IPv6 DNS server
- TACACS, RADIUS, NTP over IPv6

Restricting SNMP access to an IPv6 node

You can restrict SNMP access to a specified IPv6 host. Enter a command such as the following.

```
NetIron(config)# snmp-client ipv6 2001:efff:89::23
```

Syntax: [**no**] **snmp-client ipv6** <*ipv6-address*>

The <*ipv6-address*> must be in hexadecimal format using 16-bit values between colons, as documented in RFC 2373.

NOTE
You cannot use the following IPv6 addresses with the **snmp-client ipv6** <*ipv6-address*> command: :: (unspecified address), ff02::01 (all nodes address), and ff02:02 (all routers address)).

Specifying an IPv6 SNMP trap receiver

You can specify an IPv6 host to be a trap receiver so that all SNMP traps are sent to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. Enter a command such as the following.

```
NetIron(config)# snmp-server host ipv6 2001:efff:89::13
```

Syntax: [**no**] **snmp-server host ipv6** <*ipv6-address*>

The <*ipv6-address*> must be in hexadecimal format using 16-bit values between colons, as documented in RFC 2373.

Restricting Telnet access by specifying an IPv6 ACL

You can specify an IPv6 ACL to restrict Telnet access to management functions on the device. Enter commands similar to the following.

```
(config)# ipv6 access-list acl1
(config-ipv6-access-list acl1)# deny ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl1)# deny ipv6 30ff:3782::ff89/128 any
(config-ipv6-access-list acl1)# permit ipv6 any any
(config-ipv6-access-list acl1)# exit
(config)# telnet access-group ipv6 acl1
```

This example configures and applies an IPv6 ACL named “acl1”, which denies Telnet access to the device from the specified IPv6 addresses, but allows access from any other IPv6 address.

```
(config)# ipv6 access-list acl2
(config-ipv6-access-list acl2)# permit ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl2)# deny ipv6 any any
(config-ipv6-access-list acl2)# exit
```

This example configures and applies an IPv6 ACL named “acl2”, which allows Telnet access to the device only from the specified IPv6 address, and denies access from any other IPv6 address.

Syntax: telnet access-group ipv6 <*ipv6-acl-name*>

The <*ipv6-acl-name*> is a valid IPv6 ACL.

Restricting SSH access by specifying an IPv6 ACL

You can configure an IPv6 ACL to restrict SSH access to management functions on the device. Enter commands such as the following.

```
(config)# ipv6 access-list acl1
(config-ipv6-access-list acl1)# deny ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl1)# deny ipv6 30ff:3782::ff89/128 any
(config-ipv6-access-list acl1)# permit ipv6 any any
(config-ipv6-access-list acl1)# exit
(config)# ssh access-group ipv6 acl1
```

This example configures and applies an IPv6 ACL named “acl1”, which denies SSH access to the device from the specified IPv6 addresses, but allows access from any other IPv6 address.

```
(config)# ipv6 access-list acl2
(config-ipv6-access-list acl2)# permit ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl2)# deny ipv6 any any
(config-ipv6-access-list acl2)# exit
(config)# ssh access-group ipv6 acl2
```

This example configures and applies an IPv6 ACL named “acl2”, which allows SSH access to the device only from the specified IPv6 address, and denies access from any other IPv6 address.

Syntax: [**no**] **ssh access-group ipv6** <*ipv6-acl-name*>

The <*ipv6-acl-name*> is a valid IPv6 ACL.

Restricting Web management access by specifying an IPv6 ACL

You can configure an IPv6 ACL to restrict Web management access to management functions on the device. Enter commands such as the following.

```
(config)# ipv6 access-list acl1
(config-ipv6-access-list acl1)# deny ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl1)# deny ipv6 30ff:3782::ff89/128 any
(config-ipv6-access-list acl1)# permit ipv6 any any
(config-ipv6-access-list acl1)# exit
(config)# web access-group ipv6 acl1
```

This example configures and applies an IPv6 ACL named “acl1”, which denies Web management access to the device from the specified IPv6 addresses, but allows access from any other IPv6 address.

```
(config)# ipv6 access-list acl2
(config-ipv6-access-list acl2)# permit ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl2)# deny ipv6 any any
(config-ipv6-access-list acl2)# exit
```

This example configures and applies an IPv6 ACL named “acl2”, which allows Web management access to the device only from the specified IPv6 address, and denies access from any other IPv6 address.

Syntax: **web access-group ipv6** <*ipv6-acl-name*>

The <*ipv6-acl-name*> variable is a valid IPv6 ACL.

Restricting SNMP access by specifying an IPv6 ACL

You can configure an IPv6 ACL to restrict Web management access to management functions on the device.

NOTE
The syntax for configuring ACLs for SNMP access differs from the syntax for controlling Telnet, SSH, and Web management access using ACLs.

```
NetIron(config)# ipv6 access-list aclro
NetIron(config-ipv6-access-list aclro)# deny ipv6 host 2000:2382::e0bb:2 any
NetIron(config-ipv6-access-list aclro)# deny ipv6 30ff:3782::ff89/128 any
NetIron(config-ipv6-access-list aclro)# permit ipv6 any any
NetIron(config-ipv6-access-list aclro)# exit
NetIron(config)# ipv6 access-list aclrw
NetIron(config-ipv6-access-list aclrw)# permit ipv6 host 2000:2382::e0bb:2 any
NetIron(config-ipv6-access-list aclrw)# deny ipv6 any any
NetIron(config-ipv6-access-list aclrw)# exit
NetIron(config)# snmp-server community public ro ipv6 aclro
NetIron(config)# snmp-server community private rw ipv6 aclrw
NetIron(config)# write memory
```

These commands configure IPv6 ACLs *aclro* and *aclrw*, then apply these ACLs to community strings. ACL *aclro* controls read-only access using the “public” community string. ACL *aclrw* controls read-write access using the “private” community string.

Syntax: [**no**] **snmp-server community** <*string*> {**ro** | **rw**} ipv6 <ipv6-acl-name>

The <*string*> specifies the SNMP community string you must enter for SNMP access.

The **ro** parameter indicates that the community string is for read-only (“get”) access. The **rw** parameter indicates the community string is for read-write (“set”) access.

The **ipv6** parameter indicates that you are applying an IPv6 access list.

The <*ipv6-acl-name*> variable specifies the IPv6 access list name.

NOTE
When **snmp-server community** is configured, all incoming SNMP packets are validated first by their community strings and then by their bound ACLs. Packets are permitted if no filters are configured for an ACL.

Restricting Web management access to your device to a specific IPv6 host

You can restrict Web management access to your device to a specific IPv6 host only. Enter commands such as the following.

```
NetIron(config)# web client ipv6 3000:2383:e0bb::2/128
```

Syntax: [**no**] **web client ipv6** <*ipv6-address*>

The <*ipv6-address*> must be in hexadecimal format using 16-bit values between colons, as documented in RFC 2373.

Specifying an IPv6 Syslog server

To specify an IPv6 Syslog server, enter a command such as the following.

```
NetIron(config)# log host ipv6 2000:2383:e0bb::4/128
```

Syntax: [no] log host ipv6 <*ipv6-address*> [<*udp-port-num*>]

The <*ipv6-address*> must be in hexadecimal using 16-bit values between colons, as documented in RFC 2373.

The <*udp-port-num*> optional parameter specifies the UDP application port used for the Syslog facility.

Viewing IPv6 SNMP server addresses

Many **show** commands display IPv6 addresses for IPv6 SNMP servers. This example shows output for the **show snmp server** command.

```
NetIron# show snmp server

      Contact:
     Location:
Community(ro): .....


Traps
              Warm/Cold start: Enable
                     Link up: Enable
                   Link down: Enable
               Authentication: Enable
     Locked address violation: Enable
         Power supply failure: Enable
                  Fan failure: Enable
           Temperature warning: Enable
                 STP new root: Enable
           STP topology change: Enable
                         vsrp: Enable

 Total Trap-Receiver Entries: 4

Trap-Receiver IP-Address                Port-Number Community

     1        192.147.201.100               162     .....

     2        4000::200                     162     .....

     3        192.147.202.100               162     .....

     4        3000::200                     162     .....
```

Disabling router advertisement and solicitation messages

Router advertisement and solicitation messages enable a device to discover other devices on the same link. By default, router advertisement and solicitation message generation is enabled. To disable this feature, configure an IPv6 access list that denies them. Enter commands such as the following.

```
NetIron(config)# ipv6 access-list rtradvert
NetIron(config)# deny icmp any any router-advertisement
NetIron(config)# deny icmp any any router-solicitation
NetIron(config)# permit ipv6 any any
```

Configuring IPv4 and IPv6 protocol stacks

If a device is deployed as an endpoint for an IPv6 over IPv4 tunnel, you must configure the device to support IPv4 and IPv6 protocol stacks. Each interface that sends and receives IPv4 and IPv6 traffic must be configured with an IPv4 address and an IPv6 address. You can also explicitly enable IPv6 using the **ipv6 enable** command. Refer to "Configuring a link-local IPv6 address" on page 1719.)

To configure an interface to support both IPv4 and IPv6 protocol stacks, enter commands such as the following.

```
NetIron(config)# ipv6 unicast-routing
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ip address 192.168.1.1 255.255.255.0
NetIron(config-if-e100-3/1)# ipv6 address 2001:200:12d:1300::/64 eui-64
```

These commands globally enable IPv6 routing on the device, and configure an IPv4 address and an IPv6 address for Ethernet interface 3/1.

Syntax: [**no**] **ipv6 unicast-routing**

To disable IPv6 traffic globally on the router, enter the **no** form of this command.

Syntax: [**no**] **ip address** <*ip-address*> <*sub-net-mask*> [**secondary**]

You must specify the <*ip-address*> parameter using 8-bit values in dotted decimal notation.

You can specify the <*sub-net-mask*> parameter in either dotted decimal notation or as a decimal value preceded by a slash mark (/).

The **secondary** keyword specifies that the configured address is a secondary IPv4 address.

To remove the IPv4 address from the interface, enter the **no** form of this command.

Syntax: [**no**] **ipv6 address** <*ipv6-prefix*>/<*prefix-length*> [**eui-64**]

This syntax specifies a global or site-local IPv6 address. For information about configuring a link-local IPv6 address, refer to "Configuring a link-local IPv6 address" on page 1719.

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **eui-64** keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the MAC address of the interface. If you do not specify the **eui-64** keyword, you must manually configure the 64-bit interface ID as well as the 64-bit network prefix. For more information about manually configuring an interface ID, refer to "Configuring a global or site-local IPv6 address" on page 1718.

Configuring IPv6 Domain Name Server (DNS) resolver

The Domain Name Server (DNS) resolver feature lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a device to recognize all hosts within that domain. After you define a domain name, the device automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain "newyork.com" is defined on a device, and you want to initiate a ping to host "NYC01" on that domain, you only need to reference the host name instead of the host name and the domain name. For example, enter either of the following commands to initiate the ping.

```
NetIron# ping nyc01
NetIron# ping nyc01.newyork.com
```

Defining a DNS entry

You can define up to four DNS servers for each DNS entry. The first entry serves as the primary default address. If a query to the primary address is not resolved after three attempts, the next gateway address is queried (up to three times). This process continues for each defined gateway address until the query is resolved. The order in which the default gateway addresses are polled is the same as the order in which you enter them.

To define the domain name *newyork.com* on a device and then define four possible default DNS gateway addresses, using IPv4 addressing, enter the following commands.

```
NetIron(config)# ip dns domain-name newyork.com
NetIron(config)# ip dns server-address 209.157.22.199 205.96.7.15 208.95.7.25
201.98.7.15
```

Syntax: [**no**] **ip dns server-address** <*ip-addr*> [<*ip-addr*>] [<*ip-addr*>] [<*ip-addr*>]

In this example, the first IP address in the command becomes the primary gateway address and all others are secondary addresses. Because IP address 201.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

Defining an IPv6 DNS entry

IPv6 defines new DNS record types to resolve queries for domain names to IPv6 addresses, as well as IPv6 addresses to domain names. Devices running IPv6 software support AAAA DNS records, which are defined in RFC 1886.

AAAA DNS records are analogous to the A DNS records used with IPv4. A complete IPv6 address is stored in each record. AAAA records have a type value of 28.

To establish an IPv6 DNS entry for the device, enter the following command.

```
NetIron(config)# ipv6 dns domain-name companynet.com
```

Syntax: [**no**] **ipv6 dns domain-name** <*domain name*>

To define an IPv6 DNS server address, enter the following command.

```
NetIron(config)# ipv6 dns server-address 200::1
```

Syntax: [**no**] **ipv6 dns server-address** <*ipv6-addr*> [<*ipv6-addr*>] [<*ipv6-addr*>] [<*ipv6-addr*>]

For example, in a configuration where *ftp6.companynet.com* is a server with an IPv6 protocol stack, when a user pings *ftp6.companynet.com*, the device attempts to resolve the AAAA DNS record. In addition, if the DNS server does not have an IPv6 address, as long as it is able to resolve AAAA records, it can still respond to DNS queries.

DNS queries of IPv4 and IPv6 DNS servers

IPv4 and IPv6 DNS record queries search through IPv4 and IPv6 DNS servers as described here.

For IPv4 DNS record queries:

- Loop through all configured IPv4 DNS servers.
- If no IPv4 DNS servers are configured, then loop through all configured IPv6 DNS servers (if any).

For IPv6 DNS record queries:

- Loop through all configured IPv6 DNS servers.
- If no IPv6 DNS servers are configured, then loop through all configured IPv4 DNS servers (if any).

ECMP load sharing for IPv6

IPv6 ECMP load sharing is hardware-managed. If there is more than one path to a given destination, a hash is calculated based on the source MAC address, destination MAC address, source IPv6 address, destination IPv6 address, and TCP/UDP source port and destination port (if the packet is also a TCP and UDP packet). This hash is used to select one of the paths.

IPv6 options for load sharing are described in "Options for IP load sharing and LAGs" on page 734. Configuration of these parameters applies to IPv6 and IPv4 load sharing.

Disabling or re-enabling ECMP load sharing for IPv6

ECMP load sharing for IPv6 is enabled by default. To disable the feature, enter the following command.

```
NetIron(config)# no ipv6 load-sharing
```

To re-enable the feature after disabling it, enter the following command.

```
NetIron(config)# ipv6 load-sharing 4
```

Syntax: [**no**] **ipv6 load-sharing** [<*num*>]

The <*num*> parameter specifies the number of paths and can be from 2 – 8. When re-enabling IPv6 load sharing, you must specify a number of paths. If you do not, IPv6 load sharing will not be re-enabled.

Changing the maximum number of load sharing paths for IPv6

By default, IPv6 ECMP load sharing balances traffic across up to four equal paths. You can change the maximum number of paths to a value from 2 – 8.

To change the number of ECMP load sharing paths for IPv6, enter a command such as the following.

```
NetIron(config)# ipv6 load-sharing 8
```

Syntax: [no] ipv6 load-sharing [<*num*>]

The <*num*> parameter specifies the number of paths and can be from 2 – 8. The default is 4.

DHCP relay agent for IPv6

A client locates a DHCP server using a reserved, link-scoped multicast address. Direct communication between client and the server requires that they be attached by the same link. In some situations where ease-of-management, economy, and scalability are concerns, you can allow a DHCPv6 client to send a message to a DHCP server using a DHCPv6 relay agent. A DHCPv6 relay agent, which may reside on the client link, but is transparent to the client, relays messages between the client and the server.

When the relay agent receives a message, it creates a new relay-forward message, inserts the original DHCPv6 message, and sends the relay-forward message as the DHCP server.

Configuring DHCP for IPv6 relay agent

You can enable the DHCP for IPv6 relay agent function and specify the relay destination address (i.e. the DHCP server) on an interface by entering this command at the interface level.

```
NetIron(config)# interface ethernet 2/3
NetIron(config-if-e10000-2/3)#ipv6 dhcp-relay-dest 2001::2
```

Syntax: [no] ipv6 dhcp-relay-dest <*ipv6-address*>

Specify the <*ipv6-address*> as a destination address to which client messages are forwarded and which enables DHCP for IPv6 relay service on the interface. A maximum of 16 relay destination addresses may be entered.

Enabling support for network-based ECMP load sharing for IPv6

If network-based ECMP load sharing is configured, traffic is distributed across equal-cost paths based on the destination network address. Routes to each network are stored in CAM and accessed when a path to a network is required. Because multiple hosts are likely to reside on a network, this method uses fewer CAM entries than load sharing by host. When you configure network-based ECMP load sharing, you can choose either of the following CAM modes:

- **Dynamic mode** – In the dynamic mode, routes are entered into the CAM dynamically using a flow-based scheme, where routes are only added to the CAM as they are required. Once routes are added to the CAM, they can be aged-out when they are not in use. Because this mode conserves CAM, it is useful for situations where CAM resources are stressed or limited.
- **Static mode** – In the static mode, routes are entered into the CAM whenever they are discovered. Routes are not aged once routes are added to the CAM and can be aged-out when they are not in use.

IPv6 VPN CAM supports ECMP load sharing, which is created for IPv6 VPN routes.

Configuring the CAM mode to support network-based ECMP load sharing for IPv6

To configure the CAM mode to support network-based ECMP load sharing for IPv6, enter a command such as the following at the Global Configuration level.

```
NetIron(config)# cam-mode ipv6 dynamic
```

Syntax: [**no**] **cam-mode ipv6** [**dynamic** | **static** | **host**]

The **dynamic** parameter configures the device for network-based ECMP load sharing using the dynamic CAM mode.

The **static** parameter configures the device for network-based ECMP load sharing using the static CAM mode.

The **host** parameter configures the device for host-based ECMP load sharing using the dynamic CAM mode.

You must restart the device for this command to take effect.

Displaying ECMP load-sharing information for IPv6

To display the status of ECMP load sharing for IPv6, enter the following command.

```
NetIron# show ipv6
Global Settings
  unicast-routing enabled, ipv6 allowed to run, hop-limit 64
  reverse-path-check disabled
  urpf-exclude-default disabled
  session-logging-age 5
  No Inbound Access List Set
  No Outbound Access List Set
  Prefix-based IPv6 Load-sharing is Enabled, Number of load share paths: 4
  source-route disabled, forward-source-route disabled
Configured Static Routes: 66
Configured Static Mroutes: 66
RIP: enabled
OSPF (default VRF): enabled
BGP: enabled, 1 active neighbor(s) configured
```

Syntax: **show ipv6**

You can display the entries in the IPv6 forwarding cache by entering the **show ipv6 cache** command.

```
NetIron# show ipv6 cache
Total number of cache entries: 10
    IPv6 Address                        Next Hop                        Port
1   5000:2::2                           LOCAL                           tunnel 2
2   2000:4::106                         LOCAL                           ethe 2
3   2000:4::110                         DIRECT                          ethe 2
4   2002:c0a8:46a::1                    LOCAL                           ethe 2
5   fe80::2e0:52ff:fe99:9737            LOCAL                           ethe 2
6   fe80::ffff:ffff:feff:ffff           LOCAL                           loopback 2
7   fe80::c0a8:46a                      LOCAL                           tunnel 2
8   fe80::c0a8:46a                      LOCAL                           tunnel 6
9   2999::1                             LOCAL                           loopback 2
10  fe80::2e0:52ff:fe99:9700            LOCAL                           ethe 1
```

Syntax: **show ipv6 cache** [<*index-number*> | <*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | **ethernet** <*port*> | **ve** <*number*> | **tunnel** <*number*>]

Configuring IPv6 ICMP

ICMP for IPv6 provides error and informational messages. The stateless auto-configuration, neighbor discovery, and path MTU discovery features use ICMP messages.

This section explains how to configure the following IPv6 ICMP options:

- ICMP rate limiting
- ICMP redirects
- ICMP unreachable address or route messages

Configuring ICMP rate limiting

You can limit the rate at which IPv6 ICMP error messages are sent out on a network. For this rate-limiting implementation, IPv6 ICMP uses a token bucket algorithm.

The algorithm works using a *virtual bucket* that contains a number of tokens, where each token represents the ability to send one ICMP error message. Tokens are placed in the bucket at a specified interval until the maximum allowed number of tokens is reached. For each error message ICMP sends, a token is removed from the bucket. ICMP generates a series of error messages until the bucket is empty. When the bucket is empty, further error messages cannot be sent until a new token is placed in the bucket.

You can adjust the following elements related to the token bucket algorithm:

- The interval at which tokens are added to the bucket. The default is 100 milliseconds.
- The maximum number of tokens in the bucket. The default is 10 tokens.

For example, to adjust the interval to 1000 milliseconds and the number of tokens to 100 tokens, enter the following command.

```
NetIron(config)# ipv6 icmp error-interval 1000 100
```

Syntax: [**no**] **ipv6 icmp error-interval** <*interval*> [<*number-of-tokens*>]

The interval at which tokens are placed in the bucket has a range of 0 – 2147483647 milliseconds. The maximum number of tokens stored in the bucket has a range of 1 – 200.

NOTE
If you keep the default interval (100 milliseconds), output from the **show run** command does not show the setting of the **ipv6 icmp error-interval** command. In addition, if you configure the interval value to a number that does not evenly divide into 100000 (100 milliseconds), the system rounds the value up to the next higher value that does divide evenly. For example, if you specify an interval value of 150, the system rounds it to 200.

ICMP rate limiting is enabled by default. To disable ICMP rate limiting, set the interval to 0.

Disabling or re-enabling ICMP redirect messages

You can disable or re-enable a device to transmit ICMP redirect messages from an interface. By default, a device sends an ICMP redirect message to a neighboring host to inform it of a better first-hop device on a path to a destination. No further configuration is required to enable the sending of ICMP redirect messages. (For more information about how ICMP redirect messages are implemented for IPv6, refer to "Configuring IPv6 neighbor discovery" on page 1733.)

For example, to disable the ICMP redirect messages from Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# no ipv6 redirects
```

Syntax: [no] ipv6 redirects

To re-enable the sending of ICMP redirect messages on Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 redirects
```

Use the **show ipv6 interface** <*interface*> <*port-number*> command to verify that the sending of ICMP redirect messages is enabled on a particular interface.

Disabling ICMP error messages for source-routed IPv6 packets

By default, ICMP error messages are transmitted to announce discarded IPv6 source-routed packets that were addressed to one of the IPv6 addresses of a device. By default, these packets are discarded in software, as described in "Software filtering of IPv6 source-routed packets" on page 1740.

You can disable or re-enable the sending of ICMP error messages for discarded, IPv6 source-routed packets by using the **ipv6 icmp source-route** command. Use the **no** form of this command to disable the transmission of these error messages. The following example illustrates the disabling operation.

```
NetIron(config)# no ipv6 icmp source-route
```

Syntax: [no] ipv6 icmp source-route

Enabling ICMP error messages for an unreachable address

By default, the ICMPv6 destination unreachable messages with the code for an unreachable address are not sent for a discarded IPv6 packet. You can enable the sending of these messages by using the **ipv6 icmp unreachable address** command. This command applies globally.

For example, to enable ICMPv6 error messages for unreachable address on the current device, enter the following command.

```
NetIron(config)# ipv6 icmp unreachable address
```

Syntax: [no] ipv6 icmp unreachable address

Use the **no** parameter in front of the **ipv6 icmp unreachable address** command to disable the sending of ICMPv6 destination unreachable messages with the code is address unreachable.

Enabling ICMP messages for an unreachable route

By default, the ICMPv6 destination unreachable messages with the code for an unreachable route are not sent for a discarded IPv6 packet. You can enable the sending of these messages by using the **ipv6 icmp unreachable route** command.

For example, to enable ICMPv6 error messages for unreachable route on the current device, enter the following command.

```
NetIron(config)# ipv6 icmp unreachable route
```

Syntax: [no] ipv6 icmp unreachable route

Use the **no** parameter in front of the **ipv6 icmp unreachable route** command to disable the sending of ICMPv6 destination unreachable messages with the code for destination unreachable.

Configuring IPv6 neighbor discovery

The neighbor discovery feature for IPv6 uses IPv6 ICMP messages to do the following:

- Determine the link-layer address of a neighbor on the same link.
- Verify that a neighbor is reachable.
- Track neighbor devices.

An IPv6 host is required to listen for and recognize the following addresses, which identify this host:

- Link-local address.
- Assigned unicast address.
- Loopback address.
- All-nodes multicast address.
- Solicited-node multicast address.
- Multicast address to all other groups to which it belongs.

You can adjust the following IPv6 neighbor discovery features:

- Neighbor solicitation messages for duplicate address detection.

- Router advertisement messages:
  - Interval between router advertisement messages.
  - Value that indicates a device is advertised as a default device (for use by all nodes on a given link).
  - Prefixes advertised in router advertisement messages.
  - Flags for host stateful autoconfiguration.
- The time that an IPv6 node considers a remote node reachable (for use by all nodes on a given link).

The memory is allocated for IPv4 and IPv6 separately. The maximum IPv4 ARP and IPv6 ND entries can be supported together.

Neighbor solicitation and advertisement messages

Neighbor solicitation and advertisement messages enable a node to determine the link-layer address of another node (neighbor) on the same link. (This function is similar to the function provided by the Address Resolution Protocol [ARP] in IPv4.) For example, node 1 on a link wants to determine the link-layer address of node 2 on the same link. To do so, node 1, the source node, multicasts a neighbor solicitation message. The neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, contains the following information:

- **Source address:** IPv6 address of node 1 interface that sends the message.
- **Destination address:** solicited-node multicast address (FF02:0:0:0:0:1:FF00::/104) that corresponds the IPv6 address of node 2.
- Link-layer address of node 1.
- A query for the link-layer address of node 2.

After receiving the neighbor solicitation message from node 1, node 2 replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header. The neighbor solicitation message contains the following information:

- **Source address:** IPv6 address of the node 2 interface that sends the message.
- **Destination address:** IPv6 address of node 1.
- Link-layer address of node 2.

After node 1 receives the neighbor advertisement message from node 2, nodes 1 and 2 can now exchange packets on the link.

After the link-layer address of node 2 is determined, node 1 can send neighbor solicitation messages to node 2 to verify that it is reachable. Also, nodes 1, 2, or any other node on the same link can send a neighbor advertisement message to the all-nodes multicast address (FF02::1) if there is a change in their link-layer address.

Router advertisement and solicitation messages

Router advertisement and solicitation messages enable a node on a link to discover the devices on the same link.

Each configured interface on a link sends out a router advertisement message, which has a value of 134 in the Type field of the ICMP packet header, periodically to the all-nodes link-local multicast address (FF02::1).

A configured interface can also send a router advertisement message in response to a router solicitation message from a node on the same link. This message is sent to the unicast IPv6 address of the node that sent the router solicitation message.

At system startup, a host on a link sends a router solicitation message to the all-routers multicast address (FF01). Sending a router solicitation message, which has a value of 133 in the Type field of the ICMP packet header, enables the host to automatically configure its IPv6 address immediately instead of awaiting the next periodic router advertisement message.

Because a host at system startup typically does not have a unicast IPv6 address, the source address in the router solicitation message is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a unicast IPv6 address, the source address is the unicast IPv6 address of the host interface sending the router solicitation message.

Entering the **ipv6 unicast-routing** command automatically enables the sending of router advertisement messages on all configured interfaces. You can configure several router advertisement message parameters. For information about disabling router advertisement messages and the router advertisement parameters you can configure, refer to "Configuring reachable time for remote IPv6 nodes" on page 1738 and "Setting IPv6 router advertisement parameters" on page 1736.

Neighbor redirect messages

After forwarding a packet, by default, a device can send a neighbor redirect message to a host to inform it of a better first-hop device. The host receiving the neighbor redirect message will then readdress the packet to the better device.

A device sends a neighbor redirect message only for unicast packets, only to the originating node, and to be processed by the node.

A neighbor redirect message has a value of 137 in the Type field of the ICMP packet header.

Setting neighbor solicitation parameters for duplicate address detection

Although the stateless autoconfiguration feature assigns the 64-bit interface ID portion of an IPv6 address using the MAC address of the host's NIC, duplicate MAC addresses can occur. Therefore, the duplicate address detection feature verifies that a unicast IPv6 address is unique before it is assigned to a host interface by the stateless autoconfiguration feature. Duplicate address detection verifies that a unicast IPv6 address is unique.

If duplicate address detection identifies a duplicate unicast IPv6 address, the address is not used. If the duplicate address is the link-local address of the host interface, the interface stops processing IPv6 packets.

You can configure the following neighbor solicitation message parameters that affect duplicate address detection while it verifies that a tentative unicast IPv6 address is unique:

- The number of consecutive neighbor solicitation messages that duplicate address detection sends on an interface. By default, duplicate address detection sends three neighbor solicitation messages without any follow-up messages.
- The interval in seconds at which duplicate address detection sends a neighbor solicitation message on an interface. By default, duplicate address detection sends a neighbor solicitation message every 1 second.

NOTE
For the interval at which duplicate address detection sends a neighbor solicitation message on an interface, the device uses seconds as the unit of measure instead of milliseconds.

For example, to change the number of neighbor solicitation messages sent on Ethernet interface 3/1 to two and the interval between the transmission of the two messages to 9 seconds, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 nd dad attempt 2
NetIron(config-if-e100-3/1)# ipv6 nd ns-interval 9
```

Syntax: [no] ipv6 nd dad attempt <*number*>

Syntax: [no] ipv6 nd ns-interval <*number*>

For the number of neighbor solicitation messages, you can specify any number of attempts. Configuring a value of 0 disables duplicate address detection processing on the specified interface. To restore the number of messages to the default value, use the **no** form of this command.

For the interval between neighbor solicitation messages, you can specify any number of seconds. Not recommended for very short intervals in normal IPv6 operation. When a non-default value is configured, the configured time is both advertised and used by the device itself. To restore the default interval, use the **no** form of this command.

Setting IPv6 router advertisement parameters

You can adjust the following parameters for router advertisement messages:

- The interval (in seconds) at which an interface sends router advertisement messages. By default, an interface sends a router advertisement message every 200 seconds.
- The “router lifetime” value, which is included in router advertisements sent from a particular interface. The value (in seconds) indicates if the device is advertised as a default device on this interface. If you set the value of this parameter to 0, the device is not advertised as a default device on an interface. If you set this parameter to a value that is not 0, the device is advertised as a default device on this interface. By default, the device lifetime value included in device advertisement messages sent from an interface is 1800 seconds.

When adjusting these parameter settings, it is recommended that the interval between device advertisement transmission be less than or equal to the device lifetime value if the device is advertised as a default device. For example, to adjust the interval of device advertisements to 300 seconds and the device lifetime value to 1900 seconds on Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 nd ra-interval 300
NetIron(config-if-e100-3/1)# ipv6 nd ra-lifetime 1900
```

Syntax: [no] ipv6 nd ra-interval <*number*>

Syntax: [no] ipv6 nd ra-lifetime <*number*>

The <*number*> parameter in both commands indicates any numerical value. To restore the default interval or device lifetime value, use the **no** form of the respective command.

Controlling prefixes advertised in IPv6 router advertisement messages

By default, router advertisement messages include prefixes configured as addresses on interfaces using the **ipv6 address** command. You can use the **ipv6 nd prefix-advertisement** command to control exactly which prefixes are included in router advertisement messages. Along with which prefixes the router advertisement messages contain, you can also specify the following parameters:

- Valid lifetime — (Mandatory) The time interval (in seconds) in which the specified prefix is advertised as valid. The default is 2592000 seconds (30 days). When the timer expires, the prefix is no longer considered to be valid.
- Preferred lifetime — (Mandatory) The time interval (in seconds) in which the specified prefix is advertised as preferred. The default is 604800 seconds (7 days). When the timer expires, the prefix is no longer considered to be preferred.
- Onlink flag — (Optional) If this flag is set, the specified prefix is assigned to the link upon which it is advertised. Nodes sending traffic to addresses that contain the specified prefix consider the destination to be reachable on the local link.
- Autoconfiguration flag — (Optional) If this flag is set, the stateless auto configuration feature can use the specified prefix in the automatic configuration of 128-bit IPv6 addresses for hosts on the local link. For more information, refer to "IPv6 stateless auto-configuration" on page 1716.

For example, to advertise the prefix 2001:e077:a487:7365::/64 in router advertisement messages sent out on Ethernet interface 3/1 with a valid lifetime of 1000 seconds, a preferred lifetime of 800 seconds, and the Onlink and Autoconfig flags set, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 nd prefix-advertisement
2001:e077:a487:7365::/64 1000 800 onlink autoconfig
```

Syntax: [no] ipv6 nd prefix-advertisement <*ipv6-prefix*>/<*prefix-length*> <*valid-lifetime*> <*preferred-lifetime*> [autoconfig] [onlink]

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The valid lifetime and preferred lifetime is a numerical value between 0 – 4294967295 seconds. The default valid lifetime is 2592000 seconds (30 days), while the default preferred lifetime is 604800 seconds (7 days).

To remove a prefix from the router advertisement messages sent from a particular interface, use the **no** form of this command.

Setting flags in IPv6 router advertisement messages

An IPv6 router advertisement message can include the following flags:

- **Managed Address Configuration** — This flag indicates to hosts on a local link if they should use the stateful autoconfiguration feature to get IPv6 addresses for their interfaces. If the flag is set, the hosts use stateful autoconfiguration to get addresses as well as non-IPv6-address information. If the flag is not set, the hosts do not use stateful autoconfiguration to get addresses and if the hosts can get non-IPv6-address information from stateful autoconfiguration is determined by the setting of the Other Stateful Configuration flag.
- **Other Stateful Configuration** — This flag indicates to hosts on a local link if they can get non-IPv6 address autoconfiguration information. If the flag is set, the hosts can use stateful autoconfiguration to get non-IPv6-address information.

NOTE
When determining if hosts can use stateful autoconfiguration to get non-IPv6-address information, a set Managed Address Configuration flag overrides an unset Other Stateful Configuration flag. In this situation, the hosts can obtain non address information. However, if the Managed Address Configuration flag is not set and the Other Stateful Configuration flag is set, then the setting of the Other Stateful Configuration flag is used.

By default, the Managed Address Configuration and Other Stateful Configuration flags are not set in router advertisement messages. For example, to set these flags in router advertisement messages sent from Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 nd managed-config-flag
NetIron(config-if-e100-3/1)# ipv6 nd other-config-flag
```

Syntax: [no] ipv6 nd managed-config-flag

Syntax: [no] ipv6 nd other-config-flag

To remove either flag from router advertisement messages sent on an interface, use the **no** form of the respective command.

Configuring reachable time for remote IPv6 nodes

You can configure the duration (in seconds) that a device considers a remote IPv6 node reachable. By default, an interface uses the value of 30 seconds.

The router advertisement messages sent by an interface include the amount of time specified by the **ipv6 nd reachable-time** command so that nodes on a link use the same reachable time duration. By default, the messages include a default value of 0.

NOTE
The device uses seconds, instead of milliseconds, for the interval at which it sends router advertisement messages.

It is not recommended to configure a short reachable time duration, because a short duration causes the IPv6 network devices to process the information at a greater frequency.

For example, to configure the reachable time of 40 seconds for Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 nd reachable-time 40
```

Syntax: [no] ipv6 nd reachable-time <*seconds*>

For the <*seconds*> parameter, you can specify any numerical value. To restore the default time, use the **no** form of this command.

IPv6 source routing security enhancements

The IPv6 specification (RFC 2460) specifies support for IPv6 source-routed packets using a type 0 Routing extension header, requiring device and host to process the type 0 routing extension header. However, this requirement may leave a network open to a DoS attack.

A security enhancement disables sending IPv6 source-routed packets to IPv6 devices either completely or selectively as described in the following sections. (This enhancement conforms to RFC 5095.)

Complete filtering of IPv6 source-routed packets

Dell devices are configured to drop all IPv6 source-routed packets in hardware and software as described:

- **Hardware** – IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header are dropped in hardware by default.
- **Software** – IPv6 source-routed packets addressed to any IPv6 address on a device (regardless of where the routing extension header is located) are dropped in software by default.

Details of hardware and software filtering of IPv6 source-routed packets is provided in the following.

Hardware filtering of IPv6 source-routed packets

All IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header are automatically dropped in hardware. This filtering is performed on both IPv6 packets that require forwarding and IPv6 packets addressed to one of the IPv6 addresses on the device without sending an ICMP error message. This filtering behavior is enabled by default. Consequently, if you want a the device to process IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header you must direct it to perform this action through use of the **ipv6 forward-source-route** command, as shown in the following.

```
NetIron(config)# ipv6 forward-source-route
```

Syntax: [no] ipv6 forward-source-route

The default condition is for source-routed packets to be dropped. If you enable forwarding using this command, you can return to the default state by using the **no** option in front of the command.

NOTE
Source routed, IPv6 packets where the type 0 routing extension header does not follow directly after the IPv6 header are not automatically dropped in hardware.

Software filtering of IPv6 source-routed packets

By default, all IPv6 source-routed packets addressed to any IPv6 address on a PowerConnect are dropped by software (regardless of where the Routing Extension Header resides). You can enable the forwarding of these packets by using the **ipv6 source-route** command, as the following example shows.

```
NetIron(config)# ipv6 source-route
```

Syntax: [no] ipv6 source-route

The default condition is to disallow the forwarding of source-routed packets to IPv6 addresses. If you enable forwarding by using this command, you can return to the default state by using the **no** option of the command.

The **ipv6 forward-source-route** command must be enabled for the **ipv6 source-route** command to operate.

By default, ICMP error messages are sent for packets dropped by software. You can use the **ipv6 icmp source-route** command to disable the generation of ICMPv6 parameter problem for software discarded IPv6 source-routed packets addressed to one of the IPv6 addresses of a device. This is described in "Disabling ICMP error messages for source-routed IPv6 packets" on page 1732.

Selective filtering of IPv6 source-routed packets using ACLs

You can selectively filter IPv6 source-routed packets using ACLs. This is accomplished by creating an IPv6 ACL that specifies a type 0 routing extension header. This is done using the **routing-header-type** option when configuring an IPv6 ACL as described in "Filtering packets based on routing header type" on page 1778. An example of an IPv6 ACL that selectively drops IPv6 source-routed packets is shown in the following.

```
NetIron(config)# ipv6 access-list deny-access1
NetIron(config-ipv6-access-list deny-access1)#deny ipv6 any any
routing-header-type 0
```

As with complete filtering, selective filtering can be done in both hardware and software as described:

- **Hardware** – Inbound and outbound IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header can be selectively dropped in hardware through use of an IPv6 ACL and bound to an interface using the **ipv6 traffic-filter** command.
- **Software** – Inbound IPv6 source-routed packets that contain a routing extension header anywhere in a packet can be selectively dropped in software using an IPv6 ACL and bound to interfaces using the **ipv6 access-class** command.

Details about how to configure selective hardware and software filtering of IPv6 source-routed packets are provided in the following.

Selective hardware filtering of IPv6 source-routed packets

Both inbound and outbound IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header can be selectively dropped in hardware using an IPv6 ACL. source-routed packets dropped in hardware are dropped without an ICMP error message being sent. To apply an IPv6 ACL with the **routing-header-type** option for hardware filtering, you must apply the IPv6 ACL to specific ports using the **ipv6 traffic-filter** command as shown in the following example.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 traffic-filter deny-access1 in
```

Additionally, you must also enable forwarding using the **ipv6 forward-source-route** command (as shown in the following) to allow any forwarding of IPv6 source-routed packets.

```
NetIron(config)# ipv6 forward-source-route
```

Selective software filtering of IPv6 source-routed packets

Inbound IPv6 source-routed packets that contain a routing extension header anywhere in a packet can be selectively dropped in software using an IPv6 ACL. source-routed packets dropped in software generate ICMP Destination Unreachable error messages.

NOTE
This filtering only applies to packets addressed to one of the IPv6 addresses of the device.

To apply an IPv6 ACL with the **routing-header-type** option for software filtering, you must apply the IPv6 ACL system wide using the **ipv6 access-class** command.

```
NetIron(config)# # ipv6 access-class deny-access1 in
```

Additionally, you must also enable forwarding using the **ipv6 forward-source-route** and **ipv6 source-route** commands (as shown in the following) to allow any forwarding of IPv6 source-routed packets.

```
NetIron(config)# ipv6 forward-source-route
NetIron(config)# ipv6 source-route
```

Complete and selective filtering combination and order of application

If the complete filtering of IPv6 source-routed packets is enabled (the default state) then selective filtering cannot be performed. Consequently, you must use the **ipv6 forward-source-route** and **ipv6 source-route** commands to allow IPv6 source-routed packets when you are selectively allowing some IPv6 source-routed packets.

The following configuration of complete hardware and software filtering can be used with selective filtering if the commands are configured in the correct order:

- When the **ipv6 forward-source-route** command is configured, IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header are not dropped by hardware.
- All IPv6 source-routed packets addressed to any IPv6 address on a PowerConnect (regardless of where the Routing Extension Header is located) are dropped by software. This is the default configuration.

When using the **ipv6 forward-source-route** and **ipv6 source-route** commands as described, the filtering is performed in the order described below.

1. Inbound filtering is performed on the receiving interface using an ACL applied using the **ipv6 traffic-filter** command. This filtering is performed using hardware.
2. Complete filtering for ipv6 source route. This filtering is performed by the CPU.
3. Selective filtering using an IPv6 ACL applied on a system-wide basis using the **ipv6 access-class** command.
4. Selective filtering by hardware using an IPv6 ACL bound to an interface for outbound traffic using the **ipv6 traffic-filter** command.

Configuration examples for complete and selective filtering of source routed packets

The following examples demonstrate how to use this feature for different purposes:

- Dropping all IPv6 Source Routed Packets on all Ports
- Dropping all IPv6 Source Routed Packets on a Specified Port
- Silently Dropping all IPv6 Source Routed Packets Addressed to IPv6 Addresses
- Dropping all IPv6 Source Routed Packets Addressed to IPv6 Addresses from a Specified Source
- Allowing IPv6 Source Routed Packets from a Specified Source on a Specified Interface

Each of these examples are described in detail in the following sections.

Dropping all IPv6 source-routed packets on all ports

By default, all IPv6 source-routed packets received on all device ports are dropped.

Dropping all IPv6 source-routed packets on a specified port

The following example shows a configuration that will drop all IPv6 source-routed packets received on port 1/1 of a device.

In this example, the IPv6 ACL is configured to drop any IPv6 packet with a type 0 routing header immediately after the IPv6 header.

```
NetIron(config)# ipv6 access-list deny-access1
NetIron(config-ipv6-access-list deny-access1)# deny any any ipv6
routing-header-type 0
NetIron(config-ipv6-access-list deny-access1# permit ipv6 any any
NetIron(config-ipv6-access-list deny-access1)# exit
```

The default is for the device to drop all IPv6 source-routed packets in hardware and software. Forwarding of these packets must be explicitly enabled using the **ipv6 forward-source-route** and **ipv6 source-route** commands as shown.

```
NetIron(config)# ipv6 forward-source-route
NetIron(config)# ipv6 source-route
```

The IPv6 ACL must then be bound to the interface it is intended to filter as shown in the following example for the Ethernet 1/1 interface.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e100-1/1)# ipv6 traffic-filter deny-access1 in
```

Silently dropping all IPv6 source-routed packets sent to IPv6 addresses

The following configuration drops all IPv6 source-routed packets addressed to the IPv6 addresses on a device without sending an ICMP error message.

ICMPv6 parameter problem error messages are sent for dropped IPv6 source-routed packets addressed to the IPv6 addresses on the device. To disable these messages, use the **no** option with the **ipv6 icmp source-route** command.

```
NetIron(config)# no ipv6 icmp source-route
```

By default, the device drops all IPv6 source-routed packets in hardware and software. Use the **ipv6 forward-source- route** command to enable the forwarding of IPv6 source-routed packets with a type 0 routing extension header immediately after the IPv6 header, as shown in this example.

```
NetIron(config)# ipv6 forward-source-route
```

Dropping all IPv6 source-routed packets to IPv6 addresses from a specified source

This configuration demonstrates how to drop all IPv6 source-routed packets sent from a specified IPv6 address.

In this example, IPv6 ACL is configured to deny IPv6 source-routed packets with a destination address of aa:bb::1, and permit any other IPv6 packets.

```
NetIron(config)# ipv6 access-list deny-access2
NetIron(config-ipv6-access-list deny-access2)# deny host aa:bb::1 any
routing-header-type 0
NetIron(config-ipv6-access-list deny-access2)# permit ipv6 any any
NetIron(config-ipv6-access-list deny-access2)# exit
```

The IPv6 ACL is then applied globally to the device for inbound traffic using the **ipv6 access-class** command as shown.

```
NetIron(config)#ipv6 access-class deny-access2 in
```

By default, the device drops all IPv6 source-routed packets in hardware and software. Use the **ipv6 forward-source- route** and ipv6 source-route commands to enable forwarding of IPv6 source-routed packets, as shown.

```
NetIron(config)# ipv6 forward-source-route
NetIron(config)# ipv6 source-route
```

Allowing IPv6 source-routed packets from a specified source on a specified interface

The following configuration allows IPv6 source-routed packets sent from a specified source and addressed to the IPv6 address on the device to be received on port 1/1. Source-routed packets received on all other ports are denied.

NOTE
This configuration only works when the routing header type 0 appears immediately after the IPv6 header.

The following IPv6 ACL is configured to permit IPv6 source route packets that have a source address of aa:bb::1, deny any IPv6 source route packets with any other source address, and permit all other IPv6 packets.

```
NetIron(config)# ipv6 access-list allow-access
NetIron(config-ipv6-access-list allow-access)# permit ipv6 host aa:bb::1 any
routing-header-type 0
NetIron(config-ipv6-access-list allow-access)# deny any any routing-header-type 0
NetIron(config-ipv6-access-list allow-access)# permit ipv6 any any
NetIron(config-ipv6-access-list allow-access)# exit
```

Because this example permits IPv6 source-routed packets on a single specified interface, they must be explicitly dropped on all other interfaces on the router. The following IPv6 ACL is configured drop all source-routed packets.

```
NetIron(config)# ipv6 access-list drop-access
NetIron(config-ipv6-access-list drop-access)# deny any any routing-header-type 0
NetIron(config-ipv6-access-list drop-access)# permit ipv6 any any
NetIron(config-ipv6-access-list drop-access)# exit
```

The IPv6 ACL “allow-access” is bound to interface 1/1 where the IPv6 source-routed packets are permitted.

```
NetIron(config)# interface ethernet 1/1
NetIron(config-if-e100-1/1)# ipv6 traffic-filter allow-access
NetIron(config-if-e100-1/1)#exit
```

The IPv6 ACL “drop-access” is bound to all other interfaces on the device to drop IPv6 source-routed packets. The next example shows the “drop-access” ACL being bound to interface 1/2.

```
NetIron(config)# interface ethernet 1/2
NetIron(config-if-e100-1/2)# ipv6 traffic-filter drop-access in
NetIron(config-if-e100-1/2)#exit
...
```

The IPv6 ACL “drop-access” must be bound to all other interfaces on the device.

By default, the device drops all IPv6 source-routed packets in hardware and software as described in “Complete filtering of IPv6 source-routed packets” on page 1739. Use the **ipv6 forward-source-route** and **ipv6 source-route** commands to enable forwarding of IPv6 source-routed packets.

```
NetIron(config)# ipv6 forward-source-route
NetIron(config)# ipv6 source-route
```

Changing the IPv6 MTU

The IPv6 MTU is the maximum length of an IPv6 packet that can be transmitted on a particular interface. If an IPv6 packet is longer than an MTU, the host that originated the packet breaks the packet into fragments and transmits the fragments in multiple packets that are shorter than the configured MTU. You can configure the MTU on individual interfaces. Per RFC 2460, the minimum IPv6 MTU for any interface is 1280 bytes.

To configure the MTU on interface 3/1 to 1280 bytes, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 mtu 1280
```

Syntax: [**no**] **ipv6 mtu** <*bytes*>

You can specify between 1284 – (**default-max-frame-size** minus 18). If a non-default value is configured for an interface, router advertisements include an MTU option. The minimum values you can configure are: 1298 (**IP6_MIN_MTU** + 18) for Ethernet ports and 1284 (**IP6_MIN_MTU** + 4) for POS ports.

You can configure IPv6 MTU for to be greater than 1500 bytes, although the default remains at 1500 bytes.

At the global CLI level, the **ipv6 mtu** command has been deprecated and replaced by the **ipv6 global-mtu** command. For devices running release 04.1.00 or later, the old command syntax is no longer accepted except at system bootup.

To define IPv6 MTU globally, enter.

```
NetIron(config)#ipv6 global-mtu 1300
```

If the old command syntax is entered, the following error message is displayed.

```
NetIron(config)#ipv6 mtu 1500
ERROR - Command deprecated. Keyword mtu is replaced by global-mtu.
```

Syntax: [**no**] **ipv6 global-mtu** <*value*>

To define IPv6 MTU on an interface, enter.

```
NetIron(config-if-e1000-2/1)#ipv6 mtu
```

Syntax: ipv6 mtu <*value*>

NOTE
If the size of a jumbo packet received on a port is equal to the maximum frame size of - 18 (Layer 2 MAC header + CRC) and if this value is greater than the IPv4/IPv6 MTU of the outgoing port, it will be forwarded in the CPU.

How to determine the actual IPv6 MTU value

An IPv6 port can obtain an MTU value from any of the following sources:

- Default IP MTU setting
- Global MTU Setting
- Interface MTU Setting

An interface determines the actual MTU value through these processes.

1. If an IPv6 interface MTU value is configured, that value is used.
2. If an IPv6 interface MTU value is not configured and an IPv6 global MTU value is configured, the configured global MTU value is used.
3. If neither an IPv6 interface MTU value or an IPv6 global MTU value are configured, the default IPv6 MTU value of 1500 is used.

Configuring static neighbor entries

In some cases, a neighbor cannot be reached using neighbor discovery. In this situation, you can add a static entry to the IPv6 neighbor discovery cache, which causes a neighbor to be reachable at all times without using neighbor discovery. (A static entry in the IPv6 neighbor discovery cache functions like a static ARP entry in IPv4.)

For example, to add a static entry for a neighbor with the IPv6 address 3001:ffe0:2678:47b and link-layer address 0004.6a2b.8641 that is reachable through Ethernet interface 3/1, enter the following command.

```
NetIron(config)# ipv6 neighbor 3001:ffe0:2678:47b ethernet 3/1 0004.6a2b.8641
```

Syntax: [**no**] **ipv6 neighbor** <*ipv6-address*> **ethernet** <*port*> | **ve** <*ve-number*> [**ethernet** <*port*>] <*link-layer-address*>

The <*ipv6-address*> parameter specifies the address of the neighbor.

The **ethernet | ve** parameter specifies the interface through which to reach a neighbor. If you specify an Ethernet interface, you must also specify the port number. If you specify a VE, specify the VE number and then the Ethernet port numbers associated with the VE. The link-layer address is a 48-bit hardware address of the neighbor.

If you attempt to add an entry that already exists in the neighbor discovery cache, the software changes the already existing entry to a static entry.

To remove a static IPv6 entry from the IPv6 neighbor discovery cache, use the **no** form of this command.

Limiting the number of hops an IPv6 packet can traverse

By default, the maximum number of hops an IPv6 packet can traverse is 64. You can change this value to between 1 – 255 hops. For example, to change the maximum number of hops to 70, enter the following command.

```
NetIron(config)# ipv6 hop-limit 70
```

Syntax: [**no**] **ipv6 hop-limit** <*number*>

The number of hops can be from 1 – 255.

QoS for IPv6 traffic

Configuring QoS for IPv6 traffic is generally the same as it is for IPv4 traffic. The QoS policies you configure on the device apply to both incoming IPv6 and IPv4 traffic. ACLs can be used to perform QoS for IPv6 traffic for the following values as described in 21, “Access Control List”:

- dscp
- fragments
- priority-force
- priority-mapping
- source routing
- drop-precedence
- drop-precedence force

To enable QoS for IPv6 traffic, enter the following commands.

```
NetIron(config)# port-priority
NetIron(config)# write memory
NetIron(config)# end
NetIron# reload
```

Syntax: [**no**] **port-priority**

NOTE
You must save the configuration and reload the software to place the change into effect. This applies whether you are enabling QoS for IPv6 or IPv4 traffic.

The **port-priority** command globally enables QoS for IPv6 traffic on all interfaces. On Dell devices, when QoS is enabled with the **port-priority** command, the device inserts a value in the internal header based on a combination of the following information:

- 802.1p priority
- Interface priority (if configured)
- VLAN priority (if configured)
- The DSCP field in the Type of Service (ToS) header

For more information, refer to the 9, “Configuring Quality of Service for the NetIron MLX”.

Clearing global IPv6 information

You can clear the following global IPv6 information:

- Entries from the IPv6 cache.
- Entries from the IPv6 neighbor table.
- IPv6 routes from the IPv6 route table.
- IPv6 traffic statistics.
- IPv6 session flows

Clearing the IPv6 cache

You can remove all entries from the IPv6 cache or specify an entry based on the following:

- IPv6 prefix.
- IPv6 address.
- Interface type.

For example, to remove entries for IPv6 address 2000:e0ff::1, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 cache 2000:e0ff::1
```

Syntax: **clear ipv6 cache** [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | **ethernet** <*port*> | **pos** <*number*> |**tunnel** <*number*> | **ve** <*number*>] [**vrf** <*vrf-name*>]

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet** | **pos** | **tunnel** | **ve** parameter specifies the interfaces for which you can remove cache entries. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number, respectively.

The <*vrf-name*> parameter specifies the VRF for which you want to clear the cache entry. If no vrf parameter is entered, the default VRF is used.

Clearing IPv6 neighbor information

You can remove all entries from the IPv6 neighbor table or specify an entry based on the following:

- IPv6 prefix.
- IPv6 address.
- Interface type.

For example, to remove entries for Ethernet interface 3/1, enter the following command at the Privileged EXEC level or any of the CONFIG levels of the CLI.

```
NetIron# clear ipv6 neighbor ethernet 3/1
```

Syntax: clear ipv6 neighbor [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | **ethernet** <*port*> | **ve** <*number*>]

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet** | **ve** parameter specifies the interfaces for which you can remove cache entries. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE, you must also specify the VE number.

Clearing IPv6 routes from the IPv6 route table

You can clear all IPv6 routes or only those routes associated with a particular IPv6 prefix from the IPv6 route table and reset the routes.

For example, to clear IPv6 routes associated with the prefix 2000:7838::/32, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 route 2000:7838::/32
```

Syntax: clear ipv6 route [<*ipv6-prefix*>/<*prefix-length*>]

The <*ipv6-prefix*>/<*prefix-length*> parameter clears routes associated with a particular IPv6 prefix. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

Clearing IPv6 traffic statistics

To clear all IPv6 traffic statistics (reset all fields to zero), enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron(config)# clear ipv6 traffic
```

Syntax: clear ipv6 traffic

Deleting IPv6 session flows

To delete all flows from the IPv6 session cache, enter the following command.

```
NetIron# clear ipv6 flows
```

Syntax: clear ipv6 flows

Displaying global IPv6 information

You can display output for the following global IPv6 parameters:

- IPv6 cache.
- IPv6 interfaces.
- IPv6 neighbors.
- IPv6 route table.
- Local IPv6 routers.
- IPv6 TCP connections and the status of individual connections.
- IPv6 traffic statistics.
- IPv6 session flows

Displaying IPv6 cache information

The IPv6 cache contains an IPv6 host table with indices to the next hop gateway and the interface on which the route was learned.

To display IPv6 cache information, enter the following command at any CLI level.

```
NetIron# show ipv6 cache
Total number of cache entries: 10
    IPv6 Address                    Next Hop                    Port
1   5000:2::2                       LOCAL                       tunnel 2
2   2000:4::106                     LOCAL                       ethe 3/2
3   2000:4::110                     DIRECT                      ethe 3/2
4   2002:c0a8:46a::1                LOCAL                       ethe 3/2
5   fe80::2e0:52ff:fe99:9737        LOCAL                       ethe 3/2
6   fe80::ffff:ffff:feff:ffff       LOCAL                       loopback 2
7   fe80::c0a8:46a                  LOCAL                       tunnel 2
8   fe80::c0a8:46a                  LOCAL                       tunnel 6
9   2999::1                         LOCAL                       loopback 2
10  fe80::2e0:52ff:fe99:9700        LOCAL                       ethe 3/1
```

Syntax: show ipv6 cache [<*index-number*> | <*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | **ethernet** <*port*> | **pos** <*number*>| **ve** <*number*> | **tunnel** <*number*>][**vrf** <*vrf-name*>]

The <*index-number*> parameter restricts the display to the entry for the specified index number and subsequent entries.

The <*ipv6-prefix*>/<*prefix-length*> parameter restricts the display to the entries for the specified IPv6 prefix. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **ethernet | pos | ve | tunnel** parameter restricts the display to the entries for the specified interface. The <*ipv6-address*> parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **vrf** <vrf-name> parameter specifies the VRF for which you want to display the cache entry. If a vrf parameter is not entered, then the default VRF is used.

If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE interface, you must also specify the VE number. If you specify a tunnel interface, you must also specify the tunnel number.

This display shows the following information.

TABLE 301 IPv6 cache information fields

This field...	Displays...
Total number of cache entries	The number of entries in the cache table.
IPv6 Address	The host IPv6 address.
Next Hop	The next hop, which can be one of the following: • Direct – The next hop is directly connected to the device. • Local – The next hop is originated on this device. • <*ipv6 address*> – The IPv6 address of the next hop.
Port	The port on which the entry was learned.

Displaying IPv6 interface information

To display IPv6 interface information, enter the following command at any CLI level.

```
NetIron# show ipv6 interface
Routing Protocols : R - RIP  O - OSPF I - ISIS
Interface      Status     Routing  Global Unicast Address
Ethernet 3/3   down/down  R
Ethernet 3/5   down/down
Ethernet 3/17  up/up               2017::c017:101/64
Ethernet 3/19  up/up               2019::c019:101/64
VE 4           down/down
VE 14          up/up               2024::c060:101/64
Loopback 1     up/up               ::1/128
Loopback 2     up/up               2005::303:303/128
Loopback 3     up/up
```

Syntax: show ipv6 interface [<*interface*> [<*port-number*> |<*number*>]]

The <*interface*> parameter displays detailed information for a specified interface. For the interface, you can specify the **Ethernet**, **loopback**, **tunnel**, or **VE** keywords. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, you must also specify the number associated with the interface.

This display shows the following information.

TABLE 302 General IPv6 interface information fields

This field...	Displays...
Routing protocols	A one-letter code that represents a routing protocol that can be enabled on an interface.
Interface	The interface type, and the port number or number of the interface.
Status	The status of the interface. The entry in the Status field will be either “up/up” or “down/down”.
Routing	The routing protocols enabled on the interface.
Global Unicast Address	The global unicast address of the interface.

Displaying IPv6 interface information for a specified interface

To display detailed information for a specific interface, enter a command such as the following at any CLI level.

```
NetIron# show ipv6 interface ethernet 3/1
Interface Ethernet 3/1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::2e0:52ff:fe99:97
  Global unicast address(es):
  Joined group address(es):
    ff02::9
    ff02::1:ff99:9700
    ff02::2
    ff02::1
  MTU is 1500 bytes
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30 seconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1 seconds
  ND advertised retransmit interval is 0 seconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  No Inbound Access List Set
  No Outbound Access List Set
  RIP enabled
  RxPkts:  200  TxPkts:  200
  RxBytes: 12000 TxBytes: 12000
```

This display shows the following information.

TABLE 303 Detailed IPv6 interface information fields

This field...	Displays...
Interface/line protocol status	The status of interface and line protocol. If you have disabled the interface with the **disable** command, the status will be "administratively down". Otherwise, the status is either "up" or "down".
IPv6 status/link-local address	The status of IPv6. The status is either "enabled" or "disabled". Displays the link-local address, if one is configured for the interface.
Global unicast address(es)	Displays the global unicast addresses, if one or more are configured for the interface.
Joined group address(es)	The multicast addresses that a device interface listens for and recognizes.
MTU	The setting of the maximum transmission unit (MTU) configured for the IPv6 interface. The MTU is the maximum length an IPv6 packet can have to be transmitted on the interface. If an IPv6 packet is longer than an MTU, the host that originated the packet fragments the packet and transmits its contents in multiple packets that are shorter than the configured MTU.
ICMP	The setting of the ICMP redirect parameter for the interface.
ND	The setting of the various neighbor discovery parameters for the interface.
Access List	The inbound and outbound access lists applied to the interface.
Routing protocols	The routing protocols enabled on the interface.
RxPkts	The number of packets received at the specified port. This field supports IPv4 and IPv6 packet and byte counters.
TxPkts	The number of packets transmitted from the specified port. This field supports IPv4 and IPv6 packet and byte counters.
RxBytes	The number of bytes received at the specified port. This field supports IPv4 and IPv6 packet and byte counters.s
TxBytes	The number of bytes transmitted from the specified port. This field supports IPv4 and IPv6 packet and byte counters.

Displaying interface counters for all ports

Previous versions of the Multi-Service IronWare software support IPv4 and IPv6 packet and byte counters. The contents of these counters can be displayed for all ports on a device or per port. Output from the **show ipv6 interface ethernet** command includes packet and byte counter information on a per-port basis. Refer to "Displaying IPv6 interface information for a specified interface" on page 1751.

The default byte counters include the 20-byte per-packet Ethernet overhead. You can configure a device to exclude the 20-byte per-packet Ethernet overhead from byte accounting using the **vlan-counter exclude-overhead** command. Refer to "Extended VLAN counters for 8x10G modules" on page 265.

IPv4 and IPv6 commands display the interface counters for all ports on a router. The following example displays packet and byte counter information for all ports.

```
NetIron# show ipv6 interface counters
Interface      RxPkts     TxPkts     RxBytes     TxBytes
eth 3/3        200        200        850000      850000
eth 3/4        500        500        40000       40000
```

Syntax: show ipv6 interface counters

Table 304 describes the fields that display interface counter statistics.

TABLE 304 Interface counter display statistics

This field...	Displays...
Interface	The interface for which counter statistics are being displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Clearing the interface counters

Use the following command to clear all interface counters on a router.

```
NetIron# clear ip interface counters
```

Syntax: clear ip interface counters

Use the following command to clear the interface counters for a specified port.

```
NetIron# clear ip interface ethernet 3/2
```

Syntax: clear ip interface ethernet <*port-number*>

The **port-number** variable specifies the slot and port number for which you want to clear the interface counters.

Displaying IPv6 neighbor information

You can display the IPv6 neighbor table, which contains an entry for each IPv6 neighbor with which the device exchanges IPv6 packets.

To display the IPv6 neighbor table, enter the following command at any CLI level.

```
NetIron(config)# show ipv6 neighbor
Total number of Neighbor entries: 3
    IPv6 Address                          LinkLayer-Addr State Age Port    IsR
1   2000:4::110                           00e0.5291.bb37 REACH 20  ethe 3/1  1
2   fe80::2e0:52ff:fe91:bb37              00e0.5291.bb37 DELAY 1   ethe 3/2  1
3   fe80::2e0:52ff:fe91:bb40              00e0.5291.bb40 STALE 5930 ethe 3/3  1
```

Syntax: show ipv6 neighbor [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | <*interface*> [<*port*> |<*number*>]]

The <*ipv6-prefix*>/<*prefix-length*> parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The <*ipv6-address*> parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*interface*> parameter restricts the display to the entries for the specified router interface. For this parameter, you can specify the **Ethernet** or **VE** keywords. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE interface, you must also specify the VE number.

This display shows the following information.

TABLE 305 IPv6 neighbor information fields

This field...	Displays...
Total number of neighbor entries	The total number of entries in the IPv6 neighbor table.
IPv6 Address	The 128-bit IPv6 address of the neighbor.
Link-Layer Address	The 48-bit interface ID of the neighbor.
State	The current state of the neighbor. Possible states are as follows: • INCOMPLETE – Address resolution of the entry is being performed. • REACH – The forward path to the neighbor is functioning properly. • STALE – This entry has remained unused for the maximum interval. While stale, no action takes place until a packet is sent. • DELAY – This entry has remained unused for the maximum interval, and a packet was sent before another interval elapsed. • PROBE – Neighbor solicitation are transmitted until a reachability confirmation is received.
Age	The number of seconds the entry has remained unused. If this value remains unused for the number of seconds specified by the **ipv6 nd reachable-time command** (the default is 30 seconds), the entry is removed from the table.
Port	The port on which the entry was learned.
IsR	Determines if the neighbor is a device or host: 0 – Indicates that the neighbor is a host. 1 – Indicates that the neighbor is a device.

Displaying the IPv6 route table

To display the IPv6 route table, enter the following command at any CLI level.

```
NetIron# show ipv6 route
IPv6 Routing Table - 6 entries:
Type Codes -  B:BGP C:Connected I:ISIS L:Local O:OSPF R:RIP S:Static
OSPF  Type:  i - Inter, 1 - External Type1, 2 - External Type2, e - External
Type IPv6 Prefix          Next Hop Router           Interface  Dis/Metric Uptime
C  2001::/64               ::                        eth 1/7     0/0       45m18s
C  2001:470:0:25::/64      ::                        loopback 1  0/0       1h0m
L  2001:470:0:25::1/128    ::                        loopback 1  0/0       13m18s
C  5000:2000::/64          ::                        eth 1/13    0/0       1h0m
O  8000:4000::1/128        fe80::202:17ff:fe6e:c41c  eth 1/13    110/1     2m42s
O1 9000::1/128             fe80::20c:dbff:fef4:7406  eth 1/7     110/2     13m18s
```

Syntax: **show ipv6 route** [<*ipv6-address*> | <*ipv6-prefix*>/<*prefix-length*> | **bgp** | **connect** | **ospf** | **rip** | **isis** | **static** | **summary**]

The <*ipv6-address*> parameter restricts the display to the entries for the specified IPv6 address. You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*ipv6-prefix*>/<*prefix-length*> parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **bgp** keyword restricts the display to entries for BGP4+ routes.

The **connect** keyword restricts the display to entries for directly connected interface IPv6 routes.

The **isis** keyword restricts the display to entries for IPv6 IS-IS routes.

The **ospf** keyword restricts the display to entries for OSPFv3 routes.

The **rip** keyword restricts the display to entries for RIPng routes.

The **static** keyword restricts the display to entries for static IPv6 routes.

The **summary** keyword displays a summary of the prefixes and different route types.

The following table lists the information displayed by the **show ipv6 route** command.

TABLE 306 IPv6 route table fields

This field...	Displays...
Number of entries	The number of entries in the IPv6 route table.
Type	The route type, which can be one of the following: • B – The route is learned from BGP4+. • C – The destination is directly connected to the device. • I – The route is learned from IPv6 IS-IS. • L – The route is the host address of a loopback interface that is assigned an ipv6 address. • O – The route is learned from OSPFv3. • R – The route is learned from RIPng. • S – The route is a static route.
OSPF Type	• i – an internal route calculated by OSPF. • 1 – An OSPF type 1 external route. • 2 – An OSPF type 2 external route. • e – an external route calculated by OSPF.
IPv6 Prefix	The destination network of the route.
Next-Hop Router	The next-hop device.
Interface	The interface through which this device sends packets to reach the route destination.
Dis/Metric	The administrative distance and metric value of the route.

To display a summary of the IPv6 route table, enter the following command at any CLI level.

```
NetIron# show ipv6 route summary
IPv6 Routing Table - 7 entries:
  4 connected, 2 static, 0 RIP, 1 OSPF, 0 BGP
  Number of prefixes:
  /16: 1 /32: 1 /64: 3 /128: 2
```

Table 307 lists the information displayed by the **show ipv6 route summary** command.

TABLE 307 IPv6 route table summary fields

This field...	Displays...
Number of entries	The number of entries in the IPv6 route table.
Number of route types	The number of entries for each route type.
Number of prefixes	A summary of prefixes in the IPv6 route table, sorted by prefix length.

Displaying local IPv6 devices

The device can function as an IPv6 host, if you configure IPv6 addresses on the interfaces but do not enable IPv6 routing using the **ipv6 unicast-routing** command.

From the IPv6 host, you can display information about IPv6 devices to which the host is connected. The host learns about the devices through their router advertisement messages. To display information about the IPv6 devices connected to an IPv6 host, enter the following command at any CLI level.

```
NetIron# show ipv6 router
Router fe80::2e0:80ff:fe46:3431 on Ethernet 50, last update 0 min
Hops 64, Lifetime 1800 sec
Reachable time 0 msec, Retransmit time 0 msec
```

Syntax: show ipv6 router

If you configure your device to function as an IPv6 device (configure IPv6 addresses on the interfaces and enable IPv6 routing using the **ipv6 unicast-routing** command) and then enter the **show ipv6 router command**, you will receive the following output.

```
No IPv6 router in table
```

Meaningful output for this command is generated for devices configured to function as IPv6 hosts only.

This display shows the following information.

TABLE 308 IPv6 local router information fields

This field...	Displays...
Router <*ipv6 address*> on <*interface*> <*port*>	The IPv6 address for a particular interface.
Last update	The amount of elapsed time (in minutes) between the current and previous updates received from a device.
Hops	The default value that should be included in the Hop Count field of the IPv6 header for outgoing IPv6 packets. The hops value applies to the device for which you are displaying information and should be followed by IPv6 hosts attached to the device. A value of 0 indicates that the device leaves this field unspecified.
Lifetime	The amount of time (in seconds) that the device is useful as the default device.

TABLE 308 IPv6 local router information fields

This field...	Displays...
Reachable time	The amount of time (in milliseconds) that a device assumes a neighbor is reachable after receiving a reachability confirmation. The reachable time value applies to the device for which you are displaying information and should be followed by IPv6 hosts attached to the device. A value of 0 indicates that the device leaves this field unspecified.
Retransmit time	The amount of time (in milliseconds) between retransmissions of neighbor solicitation messages. The retransmit time value applies to the device for which you are displaying information and should be followed by IPv6 hosts attached to the device. A value of 0 indicates that the device leaves this field unspecified.

Displaying IPv6 TCP information

You can display the following IPv6 TCP information:

- General information about each TCP connection on the device, including the percentage of free memory for each of the internal TCP buffers.
- Detailed information about a specified TCP connection.

To display general information about each TCP connection on the device, enter the following command at any CLI level.

```
NetIron# show ipv6 tcp connections
Local IP address:port <-> Remote IP address:port TCP state
192.168.182.110:23 <->    192.168.8.186:4933     ESTABLISHED
192.168.182.110:8218 <->  192.168.182.106:179    ESTABLISHED
192.168.182.110:8039 <->  192.168.2.119:179      SYN-SENT
192.168.182.110:8159 <->  192.168.2.102:179      SYN-SENT
2000:4::110:179 <->       2000:4::106:8222       ESTABLISHED (1440)
Total 5 TCP connections

TCP MEMORY USAGE PERCENTAGE
FREE TCB = 98 percent
FREE TCP QUEUE BUFFER = 99 percent
FREE TCP SEND BUFFER = 97 percent
FREE TCP RECEIVE BUFFER = 100 percent
FREE TCP OUT OF SEQUENCE BUFFER = 100 percent
```

Syntax: show ipv6 tcp connections

This display shows the following information.

TABLE 309 General IPv6 TCP connection fields

This field...	Displays...
Local IP address:port	The IPv4 or IPv6 address and port number of the local interface over which the TCP connection occurs.
Remote IP address:port	The IPv4 or IPv6 address and port number of the remote interface over which the TCP connection occurs.

TABLE 309 General IPv6 TCP connection fields (Continued)

This field...	Displays...
TCP state	The state of the TCP connection. Possible states include the following: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
FREE TCB = <*percentage*>	The percentage of free TCP control block (TCB) space.
FREE TCB QUEUE BUFFER = <*percentage*>	The percentage of free TCB queue buffer space.
FREE TCB SEND BUFFER = <*percentage*>	The percentage of free TCB send buffer space.
FREE TCB RECEIVE BUFFER = <*percentage*>	The percentage of free TCB receive buffer space.
FREE TCB OUT OF SEQUENCE BUFFER = <*percentage*>	The percentage of free TCB out of sequence buffer space.

To display detailed information about a specified TCP connection, enter a command such as the following at any CLI level.

```
NetIron# show ipv6 tcp status 2000:4::110 179 2000:4::106 8222
TCP: TCB = 0x217fc300
TCP: 2000:4::110:179 <-> 2000:4::106:8222: state: ESTABLISHED Port: 1
  Send: initial sequence number = 242365900
  Send: first unacknowledged sequence number = 242434080
  Send: current send pointer = 242434080
  Send: next sequence number to send = 242434080
  Send: remote received window = 16384
  Send: total unacknowledged sequence number = 0
  Send: total used buffers 0
  Receive: initial incoming sequence number = 740437769
  Receive: expected incoming sequence number = 740507227
  Receive: received window = 16384
  Receive: bytes in receive queue = 0
  Receive: congestion window = 1459
```

Syntax: show ipv6 tcp status <*local-ip-address*> <*local-port-number*> <*remote-ip-address*> <*remote-port-number*>

The <*local-ip-address*> parameter can be the IPv4 or IPv6 address of the local interface over which the TCP connection is taking place.

The <*local-port-number*> parameter is the local port number over which a TCP connection is taking place.

The <*remote-ip-address*> parameter can be the IPv4 or IPv6 address of the remote interface over which the TCP connection is taking place.

The <*remote-port-number*> parameter is the local port number over which a TCP connection is taking place.

This display shows the following information.

TABLE 310 Specific IPv6 TCP connection fields

This field...	Displays...
TCB = <*location*>	The location of the TCB.
<*local-ip-address*> <*local-port-number*> <*remote-ip-address*> <*remote-port-number*> <*state*> <*port*>	This field provides a general summary of the following: • The local IPv4 or IPv6 address and port number. • The remote IPv4 or IPv6 address and port number. • The state of the TCP connection. For information on possible states, refer to Table 309 on page 1757. • The port numbers of the local interface.
Send: initial sequence number = <*number*>	The initial sequence number sent by the local device.
Send: first unacknowledged sequence number = <*number*>	The first unacknowledged sequence number sent by the local device.
Send: current send pointer = <*number*>	The current send pointer.
Send: next sequence number to send = <*number*>	The next sequence number sent by the local device.
Send: remote received window = <*number*>	The size of the remote received window.
Send: total unacknowledged sequence number = <*number*>	The total number of unacknowledged sequence numbers sent by the local device.
Send: total used buffers <*number*>	The total number of buffers used by the local device in setting up the TCP connection.

TABLE 310 Specific IPv6 TCP connection fields (Continued)

This field...	Displays...
Receive: initial incoming sequence number = <*number*>	The initial incoming sequence number received by the local device.
Receive: expected incoming sequence number = <*number*>	The incoming sequence number expected by the local device.
Receive: received window = <*number*>	The size of the local device receive window.
Receive: bytes in receive queue = <*number*>	The number of bytes in the local device receive queue.
Receive: congestion window = <*number*>	The size of the local device receive congestion window.

Displaying IPv6 traffic statistics

To display IPv6 traffic statistics, enter the following command at any CLI level.

```
NetIron# show ipv6 traffic
IP6 Statistics
  36947 received, 66818 sent, 0 forwarded, 36867 delivered, 0 rawout
  0 bad vers, 23 bad scope, 0 bad options, 0 too many hdr
  0 no route, 0 can't forward, 0 redirect sent, 0 source routed
  0 frag recv, 0 frag dropped, 0 frag timeout, 0 frag overflow
  0 reassembled, 0 fragmented, 0 ofragments, 0 can't frag
  0 too short, 0 too small, 11 not member
  0 no buffer, 66819 allocated, 21769 freed
  0 forward cache hit, 46 forward cache miss

ICMP6 Statistics
Received:
  0 dest unreach, 0 pkt too big, 0 time exceeded, 0 param prob
  2 echo req, 1 echo reply, 0 mem query, 0 mem report, 0 mem red
  0 router soli, 2393 router adv, 106 nei soli, 3700 nei adv, 0 redirect
  0 bad code, 0 too short, 0 bad checksum, 0 bad len
  0 reflect, 0 nd toomany opt, 0 badhopcount
Sent:
  0 dest unreach, 0 pkt too big, 0 time exceeded, 0 param prob
  1 echo req, 2 echo reply, 0 mem query, 0 mem report, 0 mem red
  0 router soli, 2423 router adv, 3754 nei soli, 102 nei adv, 0 redirect
  0 error, 0 can't send error, 0 too freq
Sent Errors:
  0 unreach no route, 0 admin, 0 beyond scope, 0 address, 0 no port
  0 source address policy, 0 reject route
  0 pkt too big, 0 time exceed transit, 0 time exceed reassembly
  0 param problem header, 0 nextheader, 0 option, 0 redirect, 0 unknown

UDP Statistics
  470 received, 7851 sent, 6 no port, 0 input errors

TCP Statistics
  57913 active opens, 0 passive opens, 57882 failed attempts
  159 active resets, 0 passive resets, 0 input errors
  565189 in segments, 618152 out segments, 171337 retransmission
```

Syntax: show ipv6 traffic

This display shows the following information.

TABLE 311 IPv6 traffic statistics fields

This field...	Displays...
IPv6 statistics	
received	The total number of IPv6 packets received by the device.
sent	The total number of IPv6 packets originated and sent by the device.
forwarded	The total number of IPv6 packets received by the router and forwarded to other devices.
delivered	The total number of IPv6 packets delivered to the upper layer protocol.
rawout	This information is used by Dell Technical Support.
bad vers	The number of IPv6 packets dropped by the device because the version number is not 6.
bad scope	The number of IPv6 packets dropped by the device because of a bad address scope.
bad options	The number of IPv6 packets dropped by the device because of bad options.
too many hdr	The number of IPv6 packets dropped by the device because the packets had too many headers.
no route	The number of IPv6 packets dropped by the device because there was no route.
can not forward	The number of IPv6 packets the device could not forward to another device.
redirect sent	This information is used by Dell Technical Support.
source routed	The number of IPv6 source-routed packets dropped.
frag recv	The number of fragments received by the device.
frag dropped	The number of fragments dropped by the device.
frag timeout	The number of fragment timeouts that occurred.
frag overflow	The number of fragment overflows that occurred.
reassembled	The number of fragmented IPv6 packets that the device reassembled.
fragmented	The number of IPv6 packets fragmented by the device to accommodate the MTU of this device or of another device.
ofragments	The number of output fragments generated by the device.
can not frag	The number of IPv6 packets the device could not fragment.
too short	The number of IPv6 packets dropped because they are too short.
too small	The number of IPv6 packets dropped because they do not have enough data.
not member	The number of IPv6 packets dropped because the recipient is not a member of a multicast group.
no buffer	The number of IPv6 packets dropped because there is no buffer available.
forward cache miss	The number of IPv6 packets received for which there is no corresponding cache entry.
ICMP6 statistics Some ICMP statistics apply to both Received and Sent, some apply to Received only, some apply to Sent only, and some apply to Sent Errors only.	
Applies to Received and Sent	

TABLE 311 IPv6 traffic statistics fields (Continued)

This field...	Displays...
dest unreach	The number of Destination Unreachable messages sent or received by the device.
pkt too big	The number of Packet Too Big messages sent or received by the device.
time exceeded	The number of Time Exceeded messages sent or received by the device.
param prob	The number of Parameter Problem messages sent or received by the device.
echo req	The number of Echo Request messages sent or received by the device.
echo reply	The number of Echo Reply messages sent or received by the device.
mem query	The number of Group Membership Query messages sent or received by the device.
mem report	The number of Membership Report messages sent or received by the device.
mem red	The number of Membership Reduction messages sent or received by the device.
router soli	The number of Router Solicitation messages sent or received by the device.
router adv	The number of Router Advertisement messages sent or received by the device.
nei soli	The number of Neighbor Solicitation messages sent or received by the device.
nei adv	The number of Router Advertisement messages sent or received by the device.
redirect	The number of redirect messages sent or received by the device.
Applies to Received Only	
bad code	The number of Bad Code messages received by the device.
too short	The number of Too Short messages received by the device.
bad checksum	The number of Bad Checksum messages received by the router.
bad len	The number of Bad Length messages received by the device.
nd toomany opt	The number of Neighbor Discovery Too Many Options messages received by the device.
badhopcount	The number of Bad Hop Count messages received by the device.
Applies to Sent Only	
error	The number of Error messages sent by the device.
can not send error	The number of times the device encountered errors in ICMP error messages.
too freq	The number of times the device has exceeded the frequency of sending error messages.
Applies to Sent Errors Only	
unreach no route	The number of Unreachable No Route errors sent by the device.
admin	The number of Admin errors sent by the device.
beyond scope	The number of Beyond Scope errors sent by the device.
address	The number of Address errors sent by the device.
no port	The number of No Port errors sent by the device.
pkt too big	The number of Packet Too Big errors sent by the device.

TABLE 311 IPv6 traffic statistics fields (Continued)

This field...	Displays...
source address policy	The number of ICMPv6 destination unreachable messages sent with code 5 because an IPv6 packet is dropped by an Access Control policy and the IPv6 source address of a packet matches the source address filtering policy.
reject route	The number of ICMPv6 destination unreachable messages sent code 6 because an IPv6 packet is dropped due to the destination address in the packet matching a route that has been configured to drop the packet.
time exceed transit	The number of Time Exceed Transit errors sent by the device.
time exceed reassembly	The number of Time Exceed Reassembly errors sent by the device.
param problem header	The number of Parameter Problem Header errors sent by the device.
nextheader	The number of Next Header errors sent by the device.
option	The number of Option errors sent by the device.
redirect	The number of Redirect errors sent by the device.
unknown	The number of Unknown errors sent by the device.
UDP statistics	
received	The number of UDP packets received by the device.
sent	The number of UDP packets sent by the device.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.
input errors	This information is used by Dell Technical Support.
TCP statistics	
active opens	The number of TCP connections opened by the device by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by the device in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Dell Technical Support.
active resets	The number of TCP connections the device reset by sending a TCP RESET message to the device at the other end of the connection.
passive resets	The number of TCP connections the device reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by Dell Technical Support.
in segments	The number of TCP segments received by the device.
out segments	The number of TCP segments sent by the device.
retransmission	The number of segments that the device retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.

Displaying IPv6 session flows

To display the contents of an IPv6 session cache, enter the following command.

```
NetIron# show ipv6 flows
```

Syntax: **show ipv6 flows** [<*source-ipv6-prefix*/*prefix-length*> | **any** | **host** <*source-ipv6_address*> <*destination-ipv6-prefix*/*prefix-length*> | **any** | **host** <*destination-ipv6-address*>]

If you do not specify a source or destination, all IPv6 flows are displayed.

Enter a value for <*source-ipv6-prefix*>/<*prefix-length*> or <*destination-ipv6-prefix*>/<*prefix-length*> to specify a source or destination prefix and prefix length that a flow must match to be included in the display.

Enter **any** for source or destination if a flow can have any source or any destination to be included in the display.

The **host** <*source-ipv6-address*> and **host** <*destination-ipv6-address*> parameters allow you specify a source or destination host IPv6 address that a flow must match to be included in the display.

EXAMPLES:

To show all IPv6 flows, enter the following command.

```
NetIron# show ipv6 flows
```

To show all IPv6 flows with any IPv6 source and any IPv6 destination addresses, enter the following command.

```
NetIron# show ipv6 flows any any
```

To show all IPv6 flows that match the source prefix 4000::/16 and any destination address, enter the following command.

```
NetIron# show ipv6 flows 4000::/16 any
```

To show all IPv6 flows that have any source address but only a destination address of host 5020::30, enter the following command.

```
NetIron# show ipv6 flows any host 5020::30
```

To show all IPv6 flows that have the source address of host 4050::30 and the destination address of host 5020::30, enter the following command.

```
NetIron# show ipv6 flows host 4050::30 host 5020::30
```

This output is an example of what is displayed when you enter the **show ipv6 flows** command.

```
NetIron# show ipv6 flows
ipv6 flows count: 6
A:Ack D:Deny E:Estab F:Fin P:Psh Pe:Permit R:Rst U:urg Fr:Fragment
Sr:SRouted
SourceAddress                          DestinationAddress
  Protocol SrcPort/IcmpType DestPort/IcmpCode Dscp FlowLabel  Flags     Age
3001::3                                3020::160
 icmp      128             0                 0    0          Pe        4
3001::3                                3020::160
 tcp       telnet          3456              0    0          DAR       3
3001::3                                3020::160
 tcp       telnet          3456              0    0          DAS       3
3001::3                                3020::160
 icmp      129             0                 0    0          Pe        8
3001::3                                3020::160
 tcp       3456            telnet            0    0          DAR       9
3001::3                                3020::165
 icmp      128             0                 0    0          Pe        4
```

The first line (ipv6 flows count) shows the number of flows included on the display.

The next line defines the flags used in the display.

Information for each flow on the display appears on two lines in the following sequence:

- **Source Address** – Source address of the flow.
- **Destination Address** – Destination address of the flow.
- **Protocol** – Protocol in the flow.
- **SrcPort/IcmpType** – Either the source TCP or UDP port or the ICMP type of the flow.
- **DestPort or IcmpCode** – Either the destination TCP or UDP port or the ICMP code of the flow.
- **Dscp** – DSCP value in the flow.
- **FlowLabel** – Value in the flow label field of the IPv6 packet header.
- **Flags** – Status of the flow, which can be a combination of different flag types. For example, DAR means the flow was denied (D), acknowledged (A), and reset (R).
- **Age** – Age of the flow.

NOTE
The life of an idle flow is 50 seconds.

Chapter

Configuring an IPv6 Prefix List 39

PowerConnect B-MLXe supports the following IPv6 Prefix List features:

- IPv6 Prefix List
- Displaying Prefix List Information

Configuring an IPv6 prefix list

PowerConnect devices support IPv6 prefix lists, which you can use for basic traffic filtering. You can configure up to 100 IPv6 prefix lists.

An IPv6 prefix list is composed of one or more conditional statements that pose an action (permit or deny) if a packet matches a specified prefix. In prefix lists with multiple statements, you can specify a sequence number for each statement.The specified sequence number determines the order in which the statement appears in the prefix.

You can configure an IPv6 prefix list on a global basis, then use it as input to other commands or processes, such as route aggregation, route redistribution, route distribution, route maps, and so on. When a router interface sends or receives an IPv6 packet, it applies the statements within the IPv6 prefix list in their order of appearance to the packet. As soon as a match occurs, the router takes the specified action (permit or deny the packet) and stops further comparison for that packet.

You can use permit statements in the prefix list to specify the traffic that you want to send to the other feature. If you use deny statements, the traffic specified by the deny statements is not supplied to the other feature.

To configure an IPv6 prefix list and use it as input to the RIPng **distribute-list** command, enter commands such as the following.

```
NetIron(config)# ipv6 prefix-list routesfor2001 permit 2001::/16
NetIron(config)# ipv6 router rip
NetIron(config-ripng-router)# distribute-list prefix-list routesfor2001 out
ethernet 3/1
```

These commands permit the inclusion of routes with the IPv6 prefix 2001::/16 in RIPng routing updates sent from Ethernet interface 3/1.

Syntax: [**no**] **ipv6 prefix-list** <*name*> [**seq** <*sequence-number*>] **deny** <*ipv6-prefix*>/<*prefix-length*> | **permit** <*ipv6-prefix*>/<*prefix-length*> | **description** <*string*> [**ge** <*ge-value*>] [**le** <*le-value*>]

The <*name*> parameter specifies the prefix list name. You use this name when using the prefix list as input to command or route map.

The **seq** <*seq-number*> parameter is optional and specifies the IPv6 prefix list's sequence number. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with prefix list entry 5. The router interprets the prefix list entries in numerical order, beginning with the lowest sequence number.

The **description** <*string*> parameter is a text string describing the prefix list.

The **deny** <*ipv6-prefix*>/<*prefix-length*> | **permit** <*ipv6-prefix*>/<*prefix-length*> parameters specify the action the router takes if a packet contains a route specified in this prefix list.

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The prefix list matches only on the specified prefix/prefix length unless you use the **ge** <*prefix-length*> or **le** <*prefix-length*> parameters. (See below.)

You can specify a range of prefix lengths for prefixes that are more specific than <*ipv6-prefix*>/<*prefix-length*>.

- If you specify only **ge** <*ge-value*>, then the range is from <*ge-value*> to 128.
- If you specify only **le** <*le-value*>, then the range is from <*le-value*> to the <*prefix-length*> parameter.

The <*ge-value*> or <*le-value*> you specify must meet the following condition.

prefix-length < ge-value <= le-value <= 128

If you do not specify **ge** <*ge-value*> or **le** <*le-value*>, the prefix list matches only on the exact prefix you specify with the <*ipv6-prefix*>/<*prefix-length*> parameter.

To delete the prefix list entry, use the **no** form of this command.

Displaying prefix list information

To display the IPv6 prefix lists configured on a router, enter the following command at any level of the CLI.

```
NetIron(config)# show ipv6 prefix-lists
ipv6 prefix-list routesfor2001: 1 entries
     seq 5 permit 2001::/16
```

Syntax: show ipv6 prefix-lists [<*name*>]

The <*name*> parameter restricts the display to the specified prefix list. Specify the name of the prefix list that you want to display.

Chapter

40

Configuring an IPv6 Access Control List

PowerConnect B-MLXe supports the following IPv6 Access Control List features:

- IPv6 Access Control List
- Filtering IPv6 Packets Based on DSCP Values
- Filtering IPv6 Packets Based on Routing Header Type
- Applying an IPv6 ACL to a Router Interface
- Adding a Comment to an IPv6 ACL Entry
- ACL CAM sharing for Inbound IPv6 ACLs
- IPv6 Extended ACLs
- ACL CAM sharing
- IPv6 ACL Accounting
- IPv6 ACL Logging

Dell devices support IPv6 access control lists (ACLs), which you can use for traffic filtering. You can configure up to 100 IPv6 ACLs. For details on Layer 2 ACLs, refer to 20, “Layer 2 Access Control Lists”. For details on IPv4 ACLs, refer to 21, “Access Control List”.

An IPv6 ACL is composed of one or more conditional statements that identify an action (permit or deny) if a packet matches a specified source or destination prefix. There can be up to 1024 statements per device.

In ACLs with multiple statements, you can specify a priority for each statement.The specified priority determines the order in which the statement appears in the ACL.The last statement is an implicit deny statement for all packets that do not match the previous statements in the ACL.

You can configure an IPv6 ACL on a global basis, then apply it to the incoming or outgoing IPv6 packets on specified router interfaces. You can apply only one IPv6 ACL to incoming traffic for an interface and only one IPv6 ACL to outgoing traffic on an interface. When an interface sends or receives an IPv6 packet, it applies the statements within the ACL (in their order of occurrence in the ACL) to the packet. When a match occurs, the router takes the specified action (permits or denies the packet) and stops further comparison for that packet.

For dynamic LAG creation and deletion using IPv6 ACLs, before a LAG is formed, all ports which will be part of the LAG must have the same configuration. After the LAG is removed, all ACL bindings (if any) are propagated to all of the secondary ports.

IPv6 ACLs enable traffic filtering based on the following information:

- IPv6 protocol
- Source IPv6 address
- Destination IPv6 address
- ICMP message type (if the protocol is ICMP)
- Source TCP or UDP port (if the IPv6 protocol is TCP or UDP)
- Destination TCP or UDP port (if the IPv6 protocol is TCP or UDP)

The IPv6 protocol can be one of the following well-known names, or any IPv6 protocol number from 0 – 255:

- Authentication Header (AHP)
- Encapsulating Security Payload (ESP)
- Internet Control Message Protocol (ICMP)
- Internet Protocol Version 6 (IPv6)
- Stream Control Transmission Protocol (SCTP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block Web access to a specific site by denying all TCP port 80 (HTTP) packets from a specified source IPv6 address to the IPv6 address of the site.

IPv6 ACLs also support the filtering of packets based on DSCP values.

NOTE
IPv6 ACLs are only applied to routed packets.This also includes mirror and deny-log actions.

Configuration considerations for IPv6 outbound ACLs on VPLS, VLL, and VLL-local endpoints

The following considerations apply to IPv6 outbound ACLs on VPLS, VLL, and VLL-local endpoints:

- Configure the port as a VPLS, VLL, or VLL-local endpoint and then bind the IPv6 outbound ACL to the port.
- Remove the IPv6 outbound ACL from a VPLS, VLL, or VLL-local endpoint before removing the port from the VPLS, VLL, or VLL-local instance or corresponding VLAN.
- Remove the IPv6 outbound ACL from a VPLS, VLL, or VLL-local endpoint before deleting the VPLS, VLL, or VLL-local instance or corresponding VLAN.
- If the VPLS, VLL, or VLL-local endpoint is a LAG port, you must first remove the IPv6 outbound ACL from the primary LAG port before deleting the LAG. This restriction is applicable even if you attempt to delete the lag using **force** keyword.
- If a VLL or VLL-local endpoint is a LAG port with an IPv6 outbound ACL, you must first remove the IPv6 outbound ACL from the primary LAG port before dynamically removing a port from the LAG.
- Ensure that no VPLS, VLL, or VLL-local endpoint exists with an IPv6 outbound ACL before entering the command: **no router mpls**.

This chapter contains the following sections:

- "Using IPv6 ACLs as input to other features" on page 1771
- "Configuring an IPv6 ACL" on page 1771
- "Applying an IPv6 ACL" on page 1785
- "Adding a comment to an IPv6 ACL entry" on page 1787

Using IPv6 ACLs as input to other features

You can use an IPv6 ACL to provide input to other features such as route maps and distribution lists. When you use an ACL this way, permit statements in the ACL specify traffic that you want to send to the other feature. If you use deny statements, the traffic specified by the deny statements is not supplied to the other feature.

Configuring an IPv6 ACL

To configure an IPv6 ACL, you must perform the following tasks:

- Create the ACL.
- Apply the ACL to a router interface.

The following configuration tasks are optional:

- Control access to and from a router.

Example configurations

To configure an access list that blocks all Telnet traffic received on port 1/1 from IPv6 host 2000:2382:e0bb::2, enter the following commands.

```
NetIron(config)# ipv6 access-list fdry
NetIron(config-ipv6-access-list-fdry)# deny tcp host 2000:2382:e0bb::2 any eq
telnet
NetIron(config-ipv6-access-list-fdry)# permit ipv6 any any
NetIron(config-ipv6-access-list-fdry)# exit
NetIron(config)# int eth 1/1
NetIron(config-if-1/1)# ipv6 traffic-filter fdry in
NetIron(config)# write memory
```

Here is another example of how to configure an ACL and apply it to an interface.

```
NetIron(config)# ipv6 access-list netw
NetIron(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
NetIron(config-ipv6-access-list-netw)# deny ipv6 host 2000:2383:e0ac::2 host
2000:2383:e0aa:0::24
NetIron(config-ipv6-access-list-netw)# deny udp any any
NetIron(config-ipv6-access-list-netw)# permit ipv6 any any
```

The first condition permits ICMP traffic from hosts in the 2000:2383:e0bb::x network to hosts in the 2001:3782::x network.

The second condition denies all IPv6 traffic from host 2000:2383:e0ac::2 to host 2000:2383:e0aa:0::24.

The third condition denies all UDP traffic.

The fourth condition permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL denies all incoming or outgoing IPv6 traffic on the ports to which the ACL is assigned.

The commands in the next example apply the ACL "netw" to the incoming and outgoing traffic on port 1/2 and to the incoming traffic on port 4/3.

```
NetIron(config)# int eth 1/2
NetIron(config-if-1/2)# ipv6 traffic-filter netw in
NetIron(config-if-1/2)# ipv6 traffic-filter netw out
NetIron(config-if-1/2)# exit
NetIron(config)# int eth 4/3
NetIron(config-if-4/3)# ipv6 traffic-filter netw in
NetIron(config)# write memory
```

Here is another example of an ACL.

```
NetIron(config)# ipv6 access-list rtr
NetIron(config-ipv6-access-list rtr)# deny tcp 2001:1570:21::/24
2001:1570:22::/24
NetIron(config-ipv6-access-list rtr)# deny udp any range 5 6 2001:1570:22::/24
NetIron(config-ipv6-access-list rtr)# permit ipv6 any any
NetIron(config-ipv6-access-list rtr)# write memory
```

The first condition in this ACL denies TCP traffic from the 2001:1570:21::x network to the 2001:1570:22::x network.

The second condition denies UDP packets from any source with source UDP ports in ranges 5 to 6 and with the 2001:1570:22::/24 network as a destination.

The third condition permits all packets containing source and destination addresses that are not explicitly denied by the first two conditions. Without this entry, the ACL would deny all incoming or outgoing IPv6 traffic on the ports to which you assign the ACL.

A **show running-config** command output resembles the following.

```
NetIron(config)# show running-config
ipv6 access-list rtr
deny tcp 2001:1570:21::/24 2001:1570:22::/24
 deny udp any range 5 6 2001:1570:22::/24
 permit ipv6 any any
```

A **show ipv6 access-list** command output resembles the following.

```
NetIron(config)# show ipv6 access-list rtr
ipv6 access-list rtr: 3 entries
 10: deny tcp 2001:1570:21::/24 2001:1570:22::/24
 20: deny udp any range 5 6 2001:1570:22::/24
 30: permit ipv6 any any
```

The following commands apply the ACL “rtr” to the incoming traffic on ports 2/1 and 2/2.

```
NetIron(config)# int eth 2/1
NetIron(config-if-2/1)# ipv6 traffic-filter rtr in
NetIron(config-if-2/1)# exit
NetIron(config)# int eth 2/2
NetIron(config-if-2/2)# ipv6 traffic-filter rtr in
NetIron(config)# write memory
```

Default and implicit IPv6 ACL action

The default action when no IPv6 ACLs are configured is to permit all IPv6 traffic. Once you configure an IPv6 ACL and apply it to an interface, the default action for that interface is to deny all IPv6 traffic that is not explicitly permitted on the interface. The following actions can be taken:

- To tightly control access, configure ACLs with permit entries for the access you want to permit. These ACLs implicitly deny all other access.
- To secure access in environments with many users, configure ACLs with explicit deny entries, then add an entry to permit all access to the end of each ACL. The router permits packets that are not denied by the deny entries.

Every IPv6 ACL has the following implicit conditions as the last match condition.

1. **permit icmp any any nd-na** – Allows ICMP neighbor discovery acknowledgement.
2. **permit icmp any any nd-ns** – Allows ICMP neighbor discovery solicitation.
3. **deny ipv6 any any** – Denies IPv6 traffic. You must enter a **permit ipv6 any any** as the last statement in the ACL to permit IPv6 traffic that was not denied by the previous statements.

The conditions are applied in the order shown above, with **deny ipv6 any any** as the last condition.

For example, to deny ICMP neighbor discovery acknowledgement, then permit any remaining IPv6 traffic, enter commands such as the following.

```
NetIron(config)# ipv6 access-list netw
NetIron(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
NetIron(config-ipv6-access-list-netw)# deny icmp any any nd-na
NetIron(config-ipv6-access-list-netw)# permit ipv6 any any
```

The first permit statement permits ICMP traffic from hosts in the 2000:2383:e0bb::x network to hosts in the 2001:3782::x network.

The deny statement denies ICMP neighbor discovery acknowledgement.

The last command permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL denies all incoming or outgoing IPv6 traffic on the ports to which you assigned the ACL.

Furthermore, if you add the statement **deny icmp any any** in the access list, then all neighbor discovery messages will be denied. You must explicitly enter the **permit icmp any any nd-na** and **permit icmp any any nd-ns** statements just before the **deny icmp** statement if you want the ACLs to permit neighbor discovery as in this example.

```
NetIron(config)# ipv6 access-list netw
NetIron(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
NetIron(config-ipv6-access-list-netw)# permit icmp any any nd-na
NetIron(config-ipv6-access-list-netw)# permit icmp any any nd-ns
NetIron(config-ipv6-access-list-netw)# deny icmp any any
NetIron(config-ipv6-access-list-netw)# permit ipv6 any any
```

ACL syntax

The following syntax rules apply for IPv6 ACLs.

Syntax: [**no**] **ipv6 access-list** <*acl name*>

Syntax: [**no**] **permit** | **deny** <*protocol*>
<*ipv6-source-prefix/prefix-length*> | **any** | **host** <*source-ipv6_address*>
<*ipv6-destination-prefix/prefix-length*> | **any** | **host** <*ipv6-destination-address*>
[**ipv6-operator** [<*value*>]]

The **ipv6 access-list** <*acl name*> parameter enables the IPv6 configuration level and defines the name of the IPv6 ACL. The <*acl name*> can contain up to 199 characters and numbers, but cannot begin with a number and cannot contain any spaces or quotation marks.

The **permit** keyword indicates that the ACL will permit (forward) packets that match a policy in the access list.

The **deny** keyword indicates that the ACL will deny (drop) packets that match a policy in the access list.

The <*protocol*> parameter indicates the type of IPv6 packet that is being filtered. You can specify a well-known name for some protocols with numbers less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI. IPv6 protocols include:

- **AHP** – Authentication Header
- **ESP** – Encapsulating Security Payload
- **IPv6** – Internet Protocol version 6
- **SCTP** – Stream Control Transmission Protocol

The <*ipv6-source-prefix*>/<*prefix-length*> and <*ipv6-destination-prefix*>/<*prefix-length*> parameters specify a source or destination prefix and prefix length that a packet must match for the specified deny or permit action to occur. You must specify the <*ipv6-source-prefix*> and <*ipv6-destination-prefix*> parameters in hexadecimal using 16-bit values between colons, as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **any** keyword, when specified instead of the <*ipv6-source-prefix*>/<*prefix-length*> or <*ipv6-destination-prefix*>/<*prefix-length*> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix ::/0.

The **host** <*ipv6-source-address*> and **host** <*ipv6-destination-address*> parameter lets you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all128 is implied.

The **ipv6-operator** [<*value*>] parameter allows you to further filter the packets using one of these options:

- **dscp** – The policy applies to packets that match the traffic class value in the traffic class field of the IPv6 packet header. This operator allows you to filter traffic based on TOS or IP precedence. You can specify a value from 0 – 63.
- **fragments** – The policy applies to fragmented packets that contain a non-zero fragment offset.

NOTE
This option is supported only when the <*protocol*> parameter is IPv6. This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags.

- **priority-force** – forces packet outgoing priority.

- **priority-mapping** – maps incoming packet priority.
- **routing** – applies only to IPv6 source-routed packets.
- **sequence** – The sequence parameter specifies where the conditional statement is to be added in the access list. You can add a conditional statement at particular place in an access list by specifying the entry number using the sequence keyword. You can specify a value from 1 – 4294967295.

NOTE
The following ACL features are not supported:

- ipv6-operator **flow-label**
- ipv6-operator **fragments** when any protocol other than IPv6 is specified. If you specify **tcp** or any other protocol instead of **ipv6**, the keyword **fragments** cannot be used.
- ipv6-operator **routing** when a protocol is specified. (Same limitation as for ipv6-operator **fragments)**

When creating ACLs, use the appropriate syntax described in the following sections for the protocol you are filtering.

For ICMP

Syntax: [**no**] **ipv6 access-list** <*acl name*>

Syntax: [**no**] **permit** | **deny icmp**
<*ipv6-source-prefix/prefix-length*> | **any** | **host** <*source-ipv6_address*>
<*ipv6-destination-prefix/prefix-length*> | **any** | **host** <*ipv6-destination-address*>
[**ipv6-operator** [<*value*>]]
[[<*icmp-type*>][<*icmp-code*>]] | [<*icmp-messge*>]

The icmp protocol indicates the you are filtering ICMP packets.

To specify an ICMP type, enter a value between 0–255 for the <*icmp-type*> parameter.

To specify an ICMP code, enter a value between 0–255 for the <*icmp-code*> parameter.

To specify an ICMP message, enter one of the following:

- beyond-scope
- destination-unreachable
- dscp
- echo-reply
- echo-request
- flow-label
- fragments
- header
- hop-limit
- mld-query
- mld-reduction
- mld-report
- nd-na

- nd-ns
- next-header
- no-admin
- no-route
- packet-too-big
- parameter-option
- parameter-problem
- port-unreachable
- reassembly-timeout
- renum-command
- renum-result
- renum-seq-number
- router-advertisement
- router-renumbering
- router-solicitation
- routing
- sequence
- time-exceeded
- unreachable

NOTE
If you do not specify a message type, the ACL applies to all ICMP message types.

For TCP

Syntax: [**no**] **ipv6 access-list** <*acl name*>

Syntax: [**no**] **permit** | **deny tcp**
<*ipv6-source-prefix/prefix-length*> | **any** | **host** <*source-ipv6_address*>
[<*tcp-udp-operator*> [<*source-port-number*>]]
<*ipv6-destination-prefix/prefix-length*> | **any** | **host** <*ipv6-destination-address*>
[[<*tcp-udp-operator*> [<*source-port-number*>]]
[**ipv6-operator** [<*value*>]] [**tcp-operator** [<*value*>]]

The tcp protocol indicates the you are filtering TCP packets.

The <*tcp-udp-operator*> parameter can be one of the following:

- **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**.
- **gt** – The policy applies to TCP or UDP port numbers greater than the port number, or the numeric equivalent of the port name you enter after **gt**. Enter "?" to list the port names.
- **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.

- **range** – The policy applies to all TCP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The <*source-port number*> and <*destination-port-number*> for the tcp-udp-operator are the numbers of the source and destination ports.

The **tcp-operator** [<*value*>] parameter specifies a comparison operator for the TCP port. This parameter applies only when you specify tcp as the protocol. You can enter one of the following operators:

- **established** – This operator applies only to TCP packets. If you use this operator, the policy applies to TCP packets that have the ACK (Acknowledgment) or RST (Reset) bits set on (set to "1") in the Control Bits field of the TCP packet header. The policy applies only to established TCP sessions, not to new sessions.
- **syn** – The policy applies to TCP packets with the SYN (Synchronize) bits set on (set to "1") in the Control Bits field of the TCP packet header.

For UDP

Syntax: [**no**] **ipv6 access-list** <*acl name*>

Syntax: [**no**] **permit** | **deny udp**
<*ipv6-source-prefix/prefix-length*> | **any** | **host** <*source-ipv6_address*>
[<*tcp-udp-operator*> [<*source port number*>]]
<*ipv6-destination-prefix/prefix-length*> | **any** | **host** <*ipv6-destination-address*>
[<*tcp-udp-operator*> [<*destination port number*>]]
[**ipv6-operator** [<*value*>]]

The udp protocol indicates the you are filtering UDP packets.

The <*tcp-udp-operator*> parameter can be one of the following:

- **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**.
- **gt** – The policy applies to TCP or UDP port numbers greater than the port number, or the numeric equivalent of the port name you enter after **gt**. Enter "?" to list the port names.
- **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.
- **range** – The policy applies to all UDP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.
- The <*source-port number*> and <*destination-port-number*> for the tcp-udp-operator are the numbers of the source and destination ports.

Filtering packets based on DSCP values

To filter packets based on DSCP values, enter commands such as the following.

```
NetIron(config)# ipv6 access-list netw
NetIron(config-ipv6-access-list netw) deny ipv6 any any dscp 3
```

Syntax: [**no**] **ipv6 access-list** <*name*>
deny | **permit**
<*ipv6-source-prefix*>/<*prefix-length*> | **any**
<*ipv6-destination-prefix*>/<*prefix-length*> | **any** [**sequence** <*number*>]
dscp <*dscp-value*>

Enter a value from 0 - 63 for the **dscp** <*dscp-value*> parameter to filter packets based on their DSCP value.

For more information on the syntax, refer to "ACL syntax" on page 1773.

Filtering packets based on routing header type

You can filter IPv6 packets based on their routing header type. This is of particular value when you want to filter IPv6 source-routed packets to prevent DoS attacks. These packets are type 0.

To filter IPv6 packets based on the routing header type, enter commands such as the following.

```
NetIron(config)# ipv6 access-list drop-source-routed
NetIron(config-ipv6-access-list drop-source-routed) deny ipv6 routing-header-type
0
```

Syntax: [**no**] **ipv6 access-list** <*name*>
deny | **permit**
routing-header-type <*type-value*>

Enter a value from 0 - 255 for the **routing-header-type** <*type-value*> parameter to filter packets based on their IPv6 header type value.

For more information on the syntax, refer to "ACL syntax" on page 1773.

NOTE
The **routing-header-type** option is separate and independent of the **routing** option. The **routing-header-type** and **routing** options are mutually exclusive and cannot be used in the same filter.

NOTE
For more information on configuring the **acl-mirror-port** command, refer to "ACL-based inbound mirroring" on page 148.

Extended IPv6 ACLs

Configuration considerations for extended IPv6 layer 4 ACL

The following configuration considerations apply to extended IPv6 L4 ACLs:

- There are two lookups available for ingress direction. In ingress direction, you can bind IPv6 layer 4 ACLs, IPv4 layer 4 ACLs, layer 2 ACLs, and layer 3 ACLs on the same port.
- The PowerConnect B-MLXe has one CAM lookup for inbound and outbound ACLs.
- Only one L2 or IPv6 lookup is allowed per port.
- There is only one lookup available for egress direction. When you bind outbound IPv6 L4 ACL to a port, the port does not allow L2, IPv4, or IPv6 ACL in that egress direction.
- Layer 4 ACLs filter incoming traffic based on IPv6 packet header fields. The following attributes can be added to the IPv6 packet header fields:
  - VLAN ID
  - Source IPv6 address (SIP) prefix
  - Destination IPv6 address (DIP) prefix
  - IP protocol (SPI matching is not supported for AHP or ESP
  - UDP or TCP source port
  - UDP or TCP destination port
  - TCP flags - established (RST or ACK)
  - TCP flags - SYN
  - ICMP type and code
  - DSCP value
  - IPv6 fragments
  - source routed packets
  - specific routing header type
- The following actions are available for the ingress ACL:
  - Permit
  - Deny
  - Copy-sflow
  - Drop-precedence
  - Drop-precedence-force
  - Priority-force
  - Mirror

The following actions are available for the egress ACL:

- Permit
- Deny

ACL syntax

The command syntax for the IPv6 ACLs is as follows.

Syntax: [**no**] **ipv6 access-list** <*acl name*>

Syntax: **permit** | **deny** <*protocol*>

<*ipv6-source-prefix/prefix-length*> | **any** | **host** <*source-ipv6_address*>

<*ipv6-destination-prefix/prefix-length*> | **any** | **host** <*ipv6-destination-address*>

[**ipv6-operator** [<*value*>]]If a port has an acl applied, the user must remove ACL bindings prior to creating or adding that port to a vlan or a ve interface.

[**copy-sflow**] | [**drop-precedence** <*dp-value*>] | [**drop-precedence-force** <*dp-value*>] | [**dscp** <*dscp-value*>] | [**mirror**] | [**priority-force** <*number*>] | [**sequence**]

The **ipv6 access-list** <*acl name*> parameter enables the IPv6 configuration level and defines the name of the IPv6 ACL. The <*acl name*> variable can contain up to 199 characters and numbers, but cannot begin with a number and cannot contain any spaces or quotation marks.

The **permit** keyword indicates that the ACL permits (forwards) packets that match a policy in the ACL.

The **deny** keyword indicates that the ACL denies (drops) packets that match a policy in the ACL.

The <*protocol*> parameter indicates the type of IPv6 packet you are filtering. You can specify a well-known name for some protocols with number lower than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI. IPv6 protocols include:

- **AHP** – Authentication Header
- **ESP** – Encapsulating Security Payload
- **IPv6** – Internet Protocol version 6
- **SCTP** – Stream Control Transmission Protocol

The <*ipv6-source-prefix*>/<*prefix-length*> and <*ipv6-destination-prefix*>/<*prefix-length*> parameters specify a source or destination prefix and prefix length that a packet must match for the specified deny or permit action to occur. You must specify the <*ipv6-source-prefix*> and <*ipv6-destination-prefix*> parameters in hexadecimal using 16-bit values between colons, as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **any** keyword, when specified instead of the <*ipv6-source-prefix*>/<*prefix-length*> or <*ipv6-destination-prefix*>/<*prefix-length*> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0

The **host** <*ipv6-source-address*> and **host** <*ipv6-destination-address*> parameter lets you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all 128 is implied.

The **ipv6-operator** [<*value*>] parameter allows you to further filter packets using one of the following options:

- **dscp** – Applies to packets that match the traffic class value in the traffic class field of the IPv6 packet header. Allows you to filter traffic based on TOS or IP precedence. You can specify a value from 0 through 63.
- **fragments** – Applies to fragmented packets that contain a non-zero fragment offset.

NOTE
This option is supported only when the <*protocol*> parameter is IPv6. This option is not applicable to filtering based on source or destination ports, TCP flags, and ICMP flags.

- **priority-force** – Forces packet outgoing priority.
- **routing** – Applies only to IPv6 source-routed packets.
- r**outing-header-type** – matches specific routing header.
- **sequence** – Specifies where the conditional statement is to be added in the ACL. You can add a conditional statement at particular place in an ACL by specifying the entry number using the sequence keyword. You can specify a value from 1 through 4294967295, as shown in this example.

```
PowerConnect(config)# ipv6 access-list ipv6-sip-dip-sample1
 deny 183 any 5001::/32
 deny 185 any host 6001::50b9
 permit 187 7017::/32 any copy-sflow
 permit 189 8017:abdc::/64 7001::/32 mirror
 permit tcp host 1616:1000:1000:1000:1000:1000:1000:1011 host
8800:1000:2000:2000:2000:2000:2000:2022 drop-precedence 2
 deny udp host 1717:1000:1000:1000:1000:1000:1000:1011 host
9900:2000:2000:2000:2000:2000:2000:2022 drop-precedence-force 1
 permit ahp host 202::12 host 201::101
 permit esp host 202::12 host 202::102
 permit ipv6 host 202::12 host 203::103 dscp 8
 permit sctp host aaa:1:202::12 host bbb::2
 permit ipv6 host 3003::110 any
 deny ipv6 dd17::/32 any fragments
 permit ipv6 a3b1:7551::/32 any priority-force 4
 permit ipv6 b3b1:7552::/32 any routing
 permit ipv6 any any routing-header-type 51
 deny 53 any 9001:a001::/32 sequence 10000
```

For ICMP

Syntax: [**no**] **ipv6 access-list** <*acl name*>

Syntax: **permit** | **deny icmp**
<*ipv6-source-prefix/prefix-length*> | **any** | **host** <*source-ipv6_address*>
<*ipv6-destination-prefix/prefix-length*> | **any** | **host** <*ipv6-destination-address*>
[**ipv6-operator** [<*value*>]]
[[<*icmp-type*>][<*icmp-code*>]] | [<*icmp-messge*>] | **beyond-scope** | **destination-unreachable** | **echo-reply** | **echo-request** | **header** | **hop-limit** | **mld-query** | **mld-reduction** | **mld-report** | **nd-na** | **nd-ns** | **next-header** | **no-admin** | **no-route** | **packet-too-big** | **parameter-option** | **parameter-problem** | **port-unreachable** | **reassembly-timeout** | **renum-command** | **renum-result** | **renum-seq-number** | **router-advertisement** | **router-renumbering** | **router-solicitation**] | [**copy-sflow**] | | [**drop-precedence** <*dp-value*>] | [**drop-precedence-force** <*dp-value*>] | [**dscp** <*dscp-value*>] | [**mirror**] | [**priority-force** <*number*>] | [**sequence**]

The icmp protocol indicates the you are filtering ICMP packets.

To specify an ICMP type, enter a value from 0 through 255 for the <*icmp-type*> parameter.

To specify an ICMP code, enter a value from 0 through 255 for the <*icmp-code*> parameter.

You can use these ICMP wild cards for IPv6 packet filtering.

- **destination-unreachable** – Matches all unreachable type codes.
- **time-exceeded** – Matches all timeout type codes.
- **router-renumbering** – Matches all router renumbering type codes.

To specify an ICMP message, enter one of the following options:

- beyond-scope
- destination-unreachable
- dscp
- echo-reply
- echo-request
- flow-label
- fragments
- header
- hop-limit
- mld-query
- mld-reduction
- mld-report
- nd-na
- nd-ns
- next-header
- no-admin
- no-route
- packet-too-big
- parameter-option
- parameter-problem
- port-unreachable
- reassembly-timeout
- renum-command
- renum-result
- renum-seq-number
- router-advertisement
- router-renumbering
- router-solicitation
- routing
- sequence
- time-exceeded
- unreachable

The following example shows a configuration to filter ICMP packets.

```
PowerConnect(config)# ipv6 access-list ipv6-icmp-sample2
 permit icmp any any echo-reply
 permit icmp any any echo-request
```

```
deny icmp any any unreachable
deny icmp any any time-exceeded
permit icmp any any 146 0
permit icmp any any 1
```

For TCP

Syntax: [**no**] **ipv6 access-list** <*acl name*>

Syntax: **permit** | **deny tcp**
<*ipv6-source-prefix/prefix-length*> | **any** | **host** <*source-ipv6_address*>
[<*tcp-udp-operator*> [<*source-port-number*>]]
<*ipv6-destination-prefix/prefix-length*> | **any** | **host** <*ipv6-destination-address*>
[<*tcp-udp-operator*> [<*destination-port- number*>]]
[**ipv6-operator** [<*value*>]] [**tcp-operator** [<*value*>]]
[**copy-sflow**] | [**drop-precedence** <*dp-value*>] | [**drop-precedence-force** <*dp-value*>] |
[**dscp** <*dscp-value*>] | [**eq** | **gt** | **lt** | **neq** | **range** <*port-number*>] | [**established**] | [**mirror**]
| [**priority-force** <*number*>] | [**sequence**] | [**syn**]

The tcp protocol indicates the you are filtering the TCP packets.

The <*tcp-udp-operator*> parameter can be one of the following:

- **eq** – Applies to the TCP or UDP port name or number you enter after **eq**.
- **gt** – Applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**. Enter "?" to list the port names.
- **lt** – Applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – Applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.
- **range** – Applies to all TCP port numbers between the first and second TCP or UDP port name or number you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The <*source-port number*> and <*destination-port-number*> for the tcp-udp-operator are the numbers of the source and destination ports.

The **tcp-operator** [<*value*>] parameter specifies a comparison operator for the TCP port. This parameter applies only when you specify tcp as the protocol. You can enter one of the following operators:

- **established** – Applies only to the TCP packets. If you use this operator, the policy applies to the TCP packets that have the ACK or RST bits set on (set to “1”) in the Control Bits field of the TCP packet header. Applies only to established TCP sessions, not to new sessions.
- **syn** – Applies to the TCP packets with the SYN bits set on (set to “1”) in the Control Bits field of the TCP packet header.

For UDP

Syntax: [**no**] **ipv6 access-list** <*acl name*>

Syntax: **permit** | **deny udp**
<*ipv6-source-prefix/prefix-length*> | **any** | **host** <*source-ipv6_address*> [**tcp-udp-operator**

[<*source port number*>]]
<*ipv6-destination-prefix/prefix-length*> | **any** | **host** <*ipv6-destination-address*>
[**tcp-udp-operator** [<*destination port number*>]]
[**ipv6-operator** [<*value*>]]
[**copy-sflow**] | [**drop-precedence** <*dp-value*>] | [**drop-precedence-force** <*dp-value*>] | [**dscp** <*dscp-value*>] | [**eq** | **gt** | **lt** | **neq** | **range** <*port-number*>] | [**mirror**] | [**priority-force** <*number*>] | [**sequence**]

The udp protocol indicates the you are filtering the UDP packets.

The <*tcp-udp-operator*> parameter can be one of the following:

- **eq** – Applies to the TCP or UDP port name or number you enter after **eq**.
- **gt** – Applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**. Enter "?" to list the port names.
- **lt** – Applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – Applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.
- **range** – Applies to all UDP port numbers that are between the first and second TCP or UDP port name or number you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The <*source-port number*> and <*destination-port-number*> are the numbers of the source port and destination port.

The following example is a configuration to filterTCP and UDP packet:

```
PowerConnect(config)# ipv6 access-list ipv6-tcp-udp-sample3
 permit tcp host 3003::11 gt 1023 host 3001::11 range 1024 1026
 deny udp host 3003::12 lt 1025 any neq 1024
 permit tcp 3001::/32 host 3002::11 syn
 permit udp any eq msg-auth 3000::/64
 permit tcp host 3003::11 gt 1023 host 3001::11 range 1024 1026 established
 deny tcp 3003::/64 range 1023 1025 host 3000::11
```

CAM partitioning

The size of the extended ingress IPv6 L4 key is 640 bits. The size of the standard ingress ACL key is 320 bits. In internal TCAM, different sized keys can reside next to each other in the same block. In external TCAM, blocks are allocated for ACLs, and different sized keys cannot reside in the same block. An ingress IPv6 L4 key cannot reside in the same block with other ingress ACLs.

You can configure CAM partition to have an ingress ACL into internal TCAM and an egress ACL into external TCAM. The ingress IPv6 L4 key can reside in the same TCAM with other ingress ACLs, but must reside in a different block in the external TCAM.

You can select one key per interface for the following packet types (port or VLAN).

- IPv6 packets
- IPv4 and ARP packets
- Non-IP packets

The following key types apply to layer 2 ACLs:

- Ingress L2 non-IP Key 0
- Egress L2+IPv4+L4 Key

The following keys apply to ether type IPv4, IPv6, or ARP:

- Ingress L2+IPv4/6 Key 1 -- ether type = IPv4 or IPv6
- Ingress IPv4+L4 Key 2 -- ether type = ARP
- Egress L2+IPv6 Key -- ether type = IPv6
- Egress L2+IPv4+L4 Key - ether type = ARP or IPv4

At ingress, each packet is subjected to two lookups. You can direct the system to use a different key for each lookup. Make sure that the source MAC, destination MAC, VLAN ID and ether type are the same for all layer 2 ACL fields. If layer 2 field locations are not same, you will have to create a separate TCAM entry for each layer 2 IPv6 ACL rule or packet type (IPv4, IPv6, and non-IP) combination, for the layer 2 IPv6 ACL to work on all packet types.

Applying an IPv6 ACL

To apply an IPv6 ACL, (for example “access1”), to an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 traffic-filter access1 in
```

This example applies the IPv6 ACL “access1” to incoming IPv6 packets on Ethernet interface 3/1. As a result, Ethernet interface 3/1 denies all incoming packets from the site-local prefix fec0:0:0:2::/64 and the global prefix 2001:100:1::/48 and permits all other incoming packets.

NOTE
IPv6 ACLs are supported on POS interfaces.

Syntax: [**no**] **ipv6 traffic-filter** <*ipv6-acl-name*> **in** | **out**

For the <*ipv6-acl-name*> parameter, specify the name of an IPv6 ACL created using the **ipv6 access-list** command.

The **in** keyword applies the specified IPv6 ACL to incoming IPv6 packets on the router interface.

The **out** keyword applies the specified IPv6 ACL to outgoing IPv6 packets on the router interface.

Applying an IPv6 ACL to a VRF

A VRF interface can be one physical port, a virtual interface, or a trunk port consisting of multiple physical ports. As with regular IPv6 ports, you can apply an inbound or outbound IPv6 ACL to a VRF interface to filter incoming and outgoing traffic respectively. This type of ACL is called a IPv6 VRF ACL.

Distinction between IPv6 ACLs applied to regular and VRF interfaces

IPv6 ACLs (both inbound and outbound) can only be applied at the IPv6 interface-level, which may be a physical or a virtual interface. If a physical port is a member of one or more virtual interfaces, the IPv6 ACL must be bound at the corresponding ve level (not at the physical port level). You cannot change the VLAN membership of a physical port with an IPv6 ACL.

When an IPv6 VRF is dynamically configured on an interface port, all IPv6 addresses on that interface are deleted. IPv6 ACL binding on the interface is not be cleared because IPv6 ACL programming is independent of the VRF membership of the interface.

To apply an IPv6 ACL, for example “access1”, to a vrf, enter commands such as the following.

```
NetIron(config)# vrf 20
NetIron(config-vif-20)#ipv6 traffic-filter access1 in
```

Syntax: **[no] ipv6 traffic-filter** <*ipv6-acl-name*> **in | out**

For the <*ipv6-acl-name*> parameter, specify the name of an IPv6 ACL created using the **ipv6 access-list** command.

The **in** keyword applies the specified IPv6 ACL to incoming IPv6 packets on the router interface.

The **out** keyword applies the specified IPv6 ACL to outgoing IPv6 packets on the router interface.

Controlling access to a router

You can use an IPv6 ACL to filter control incoming and outgoing connections to and from a router. To do so, you must create an ACL and then specify the sequence in which the ACL is applied to incoming or outgoing connections to the router.

For example, to permit incoming connections from remote hosts (2000:2383:e0bb::2/128 and 2000:2383:e0bb::3/128) to a router (30ff:3782::ff89/128), enter the following commands.

```
NetIron(config)# ipv6 access-list remote-hosts permit 2000:2383:e0bb::2/128
30ff:3782::ff89/128 sequence 10
NetIron(config)# ipv6 access-list remote-hosts permit 2000:2383:e0bb::3/128
30ff:3782::ff89/128 sequence 20
NetIron(config)# ipv6 access-class remote-hosts in
```

Because of the implicit deny command at the end of each IPv6 ACL, the router denies incoming connections from all other IPv6 hosts.

NOTE
The **ipv6 access-class** command is applicable only to traffic coming in or going out the management port.

Syntax: **[no] ipv6 access-list** <*name*> **deny | permit** <*ipv6-source-prefix*>/<*prefix-length*> **| any** <*ipv6-destination-prefix*>/<*prefix-length*> **| any [sequence** <*number*>**]**

The **sequence** <*number*> parameter specifies the order in which a statement appears in an IPv6 ACL and is therefore applied to a request. You can specify a value from 0 – 4294967295.

For more information on the syntax, refer to “ACL syntax” on page 1773.

Adding a comment to an IPv6 ACL entry

You can optionally add a comment to describe entries in an IPv6 ACL. The comment appears in the output of **show** commands that display ACL information.

You can add a comment by entering the **remark** command immediately preceding an ACL entry, or specify the ACL entry to which the comment applies.

For example, to enter comments for preceding an ACL entry, enter commands such as the following.

```
NetIron(config)#ipv6 access-list rtr
NetIron(config-ipv6-access-list rtr)# remark This entry permits ipv6 packets from
3002::2 to any destination
NetIron(config-ipv6-access-list rtr)# permit ipv6 host 3000::2 any
NetIron(config-ipv6-access-list rtr)# remark This entry denies udp packets from
any source to any destination
NetIron(config-ipv6-access-list rtr)# deny udp any any
NetIron(config-ipv6-access-list rtr)# remark This entry denies IPv6 packets from
any source to any destination
NetIron(config-ipv6-access-list rtr)# deny ipv6 any any
NetIron(config-ipv6-access-list rtr)# write memory
```

Syntax: [no] remark *<comment-text>*

The *<comment-text>* can be up to 256 characters in length.

To apply a comment to a specific ACL entry, specify the ACL's entry number with the **remark-entry sequence** command. Use the **show ipv6 access-list** command to list ACL entry number. Enter commands such as the following .

```
NetIron(config)# ipv6 access-list netw
NetIron(config-ipv6-access-list netw) remark-entry sequence 10 This entry permits
ipv6 packets from 3000::2 to any destination
NetIron(config-ipv6-access-list netw)# remark-entry sequence 20 This entry denies
UDP packets from any source to any destination
NetIron(config-ipv6-access-list netw)# remark-entry sequence 30 This entry denies
IPv6 packets from any source to any destination
```

Syntax: [no] remark-entry sequence *<sequence number>* *<comment-text>*

The *<sequence number>* is the line number assigned to the ACL entry. For a list of ACL entry numbers, use the **show ipv6 access-list** command.

The *<comment-text>* can be up to 256 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same command.

You can use the **show running-config** or **show ipv6 access-list** commands to display IPv6 ACLs and comments.

The following shows the comment text for the ACL named "rtr" in a show running-config display.

```
NetIron# show running-config
ipv6 access-list rtr
 remark This entry permits ipv6 packets from 3002::2 to any destination
 permit ipv6 host 3000::2 any
 remark This entry denies udp packets from any source to any destination
 deny udp any any
 remark This entry denies IPv6 packets from any source to any destination
 deny ipv6 any any
```

Syntax: show running-config

The following example shows the comment text for the ACL named "rtr" in a **show ipv6 access-list** display.

```
NetIron# show ipv6 access-list rtr
ipv6 access-list rtr: 3 entries
 10: remark This entry permits ipv6 packets from 3002::2 to any destination
 10: permit ipv6 host 3000::2 any
 20: remark This entry denies udp packets from any source to any destination
 20: deny udp any any
 30: remark This entry denies IPv6 packets from any source to any destination
 30: deny ipv6 any any
```

Syntax: show ipv6 access-list [<*access-list-name*>]

For the <*access-list-name*> parameter, specify the name of an IPv6 ACL created using the **ipv6 access-list** command.

Use the **all** keyword to display all IPv6 ACLs configured on the device.

ACL CAM sharing for inbound IPv6 ACLs

ACL CAM sharing allows you to conserve CAM by sharing it between ports that are supported by the same packet processor (PPCR). If this feature is enabled globally, you can share CAM space that is allocated for inbound ACLs between instances on ports that share the same packet processor (PPCR). For example, if you have bound- inbound ACL 101 to ports 1/1 and 1/5, the ACL is stored in a single location in CAM and used by both ports. Table 10 describes which ports share PPCRs and can participate in ACL CAM sharing.

TABLE 312 Common ports per PPCR

Module type	PPCR number	Ports supported by PPCR
20 x 1G	PPCR 1	1 - 20
4 x 10G	PPCR 1	1 - 2
	PPCR 2	3 - 4
2 x 10G	PPCR 1	1 - 2

Considerations when implementing this feature

The following consideration apply when implementing this feature:

- If you enable ACL CAM sharing, ACL statistics will be generated per-PPCR instead of per-port. If you require the statistics per-port granularity for your application, you cannot use this feature.
- This feature cannot be applied to a virtual interface.

- CAM entry matching within this feature is based on the ACL group ID.

Configuring ACL CAM sharing for IPv6 ACLs

When enabled, ACL CAM sharing for IPv6 inbound ACLs is applied across all ports in a system. To apply ACL CAM sharing for IPv6 ACLs globally on a PowerConnect router, use the following command.

```
NetIron(config)# ipv6 enable-acl-cam-sharing
```

Syntax: ipv6 enable-acl-cam-sharing

Filtering and priority manipulation based on 802.1p priority

Filtering and priority manipulation based on a packet's 801.1p priority is supported in the PowerConnect devices through the following QoS options:

- **priority-force** – Assigns packets of outgoing traffic that match the ACL to a specific hardware forwarding queue, even though the incoming packet may be assigned to another queue. Specify one of the following QoS queues:
  - 0 – qosp0
  - 1 – qosp1
  - 2 – qosp2
  - 3 – qosp3
  - 4 – qosp4
  - 5 – qosp5
  - 6 – qosp6
  - 7 – qosp7

  If a packet's 802.1p value is forced to another value by its assignment to a lower value queue, it will retain that value when it is sent out through the outbound port.

  The default behavior on previous revisions of this feature was to send the packet out with the higher of two possible values: the initial 802.1p value that the packet arrived with or the new (higher) priority that the packet has been "forced" to.
- **priority-mapping** – Matches on the packet's 802.1p value. This option does not change the packet's forwarding priority through the device or mark the packet.

Example using the priority force option

In the following IPv6 ACL example, access list acl1 assigns tcp packets with the source address specified and any destination address to the internal priority 7.

```
NetIron(config)# ipv6 access-list acl1
NetIron(config-ipv6-access-list acl1)# permit tcp 4000:1::/64 any priority-force
7
```

The **priority-force** parameter specifies one of the 8 internal priorities of the PowerConnect Router. Possible values are between 0 and 7.

Example using the priority mapping option

In the following IPv6 example, access list acl2 permits udp packets with the source address specified, any destination address and the 802.1p priority 5.

```
NetIron(config)# ipv6 access-list acl2
NetIron(config-ipv6-access-list acl2)# permit udp 4000:1::/64 any
priority-mapping 5
```

The **priority-mapping** parameter specifies one of the eight possible 802.1p priority values. Possible values are between 0 and 7.

NOTE
When the priority configured for a physical port and the 802.1p priority of an arriving packet differ, the higher of the two priorities is used.

ACL accounting

Multi-Service devices monitor the number of times an ACL is used to filter incoming or outgoing traffic on an interface. The **show ipv6 access-list accounting** command displays the number of "hits" or how many times ACL filters permitted or denied packets that matched the conditions of the filters.

NOTE
ACL accounting does not tabulate nor display the number of implicit denials by an ACL.

Counters, stored in hardware, keep track of the number of times an ACL filter is used.

The counters that are displayed on the ACL accounting report are:

- **1s** – Number of hits during the last second. This counter is updated every second.
- **1m** – Number of hits during the last minute. This counter is updated every one minute.
- **5m** – Number of hits during the last five minutes. This counter is updated every five minutes.
- **ac** – Accumulated total number of hits. This counter begins when an ACL is bound to an interface and is updated every one minute. This total is updated until it is cleared.

  The accumulated total is updated every minute. For example, a minute after an ACL is bound to a port, it receives 10 hits per second and continues to receive 10 hits per second. After one minute, the accumulated total hits is 600. After 10 minutes, there will be 6000 hits.

The counters can be cleared when the device is rebooted, when an ACL is bound to or unbound from an interface, or by entering a **clear ipv6 access-list** command.

Displaying statistics for IPv6 ACL accounting

To display statistics for IPv6 accounting, enter commands such as the following.

```
NetIron (config)# show ipv6 access-list accounting brief
Collecting IPv6 ACL accounting summary for 5/1 ...  Completed successfully.
Collecting IPv6 ACL accounting summary for 5/2 ...  Completed successfully.
IPv6 ACL Accounting Summary: (ac = accumulated since accounting started)
    Int    In ACL              Total In Hit   Out ACL             Total Out Hit
    5/1   fdry115               3551122(1s)
                              135155472(1m)
                                      0(5m)
                              135155472(ac)
    5/2                                      fdry116               3551123(1s)
                                                                 135154337(1m)
                                                                         0(5m)
                                                                 135154337(ac)
```

The display shows the following information.

This field...	Displays...
The IP multicast traffic snooping state	The first line of the display indicates whether IP multicast traffic snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Collecting ACL accounting summary for <*interface*>	Shows for which interfaces the ACL accounting information was collected and whether or not the collection was successful.
Int	The ID of the interface for which the statistics are being reported.
In ACL	The ID of the ACL used to filter the incoming traffic on the interface.
Total In Hit*	The number of hits from incoming traffic processed by all ACL entries (filters) in the ACL. A number is shown for each counter.
Out ACL	ID of the ACL used to filter the outgoing traffic on the interface.
Total Out Hit*	The number of hits from incoming traffic processed by all ACL entries (filters) in the ACL. A number is shown for each counter.
* The Total In Hit and Total Out Hit displays the total number of hits for all the ACL entries (or filters) in an ACL. For example, if an ACL has five entries and each entry processed matching conditions three times during the last minute, then the total Hits for the 1m counter is 15.	

Syntax: show ipv6 access-list accounting brief

Displaying IPv6 accounting statistics for an interface

To display statistics for an interface, enter commands such as the following.

```
NetIron (config)# show ipv6 access-list accounting eth 5/1 in
Collecting IPv6 ACL accounting for 5/1  ...  Completed successfully.
IPv6 ACL Accounting Information:
Inbound: IPv6 ACL fdry115
   10:  permit ipv6 host 4000::2 any
       Hit count: (1 sec)             3551111   (1 min)           135155472
                  (5 min)                   0   (accum)           135155472
```

The display shows the following information.

This field...	Displays...
The IP multicast traffic snooping state	The first line of the display indicates whether IP multicast traffic snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Collecting IPv6 ACL accounting for <*interface*>	Shows the interface included in the report and whether or not the collection was successful.
Outbound or Inbound ACL ID	Shows the direction of the traffic on the interface and the ID of the ACL used.
#	Shows the priority of the ACL entry, followed by the permit or deny condition defined for that ACL entry. ACL entries are displayed in order of ascending ACL filter priorities.
Hit count	Shows the number of hits for each counter.

Syntax: show ipv6 access-list accounting ethernet [<*slot*>/<*port*> | **ve** <*ve-number*> | **pos** [<*slot*>/<*port*>] **in | out**

Use **pos** <*slot*>/<*port*> to display a report for a POS interface.

Use **ethernet** <*slot*>/<*port*> to display a report for a physical interface.

Use **ve** <*ve-number*> to display a report for the ports that are included in a virtual routing interface. For example, if ports 1/2, 1/4, and 1/6 are all members of ve 2, the report includes information for all three ports.

Use the **in** parameter to display statistics for incoming traffic; **out** for outgoing traffic.

Clearing the ACL statistics

Statistics on the ACL account report can be cleared:

- When a software reload occurs
- When the ACL is bound to or unbound from an interface
- When you enter the **clear ipv6 access-list** command, as in the following example.

```
NetIron(config)# clear ipv6 access-list all
```

Syntax: clear ipv6 access-list all | ethernet <*slot*>/<*port*> | **ve** <*ve-num*> | **pos** <*slot*>/<*port*>

Enter **all** to clear all statistics for all ACLs.

Use **pos** <*slot*>/<*port*> to clear statistics for ACLs a physical port.

Use **ethernet** <*slot*>/<*port*> to clear statistics for ACLs a physical port.

Use **ve** <*ve-number*> to clear statistics for all ACLs bound to ports that are members of a virtual routing interface.

Chapter

41

Configuring IPv6 Routes

Table displays the IPv6 Routes features supported by PowerConnect B-MLXe Series.

- Static IPv6 Route
- IPv6 Static Multicast Route

This chapter describes how to configure a static IPv6 route. A ***static IPv6 route*** is a manually configured route, which creates a path between two IPv6 devices. A static IPv6 route is similar to a static IPv4 route. Static IPv6 routes have their advantages and disadvantages; for example, a static IPv6 route does not generate updates, which reduces processing time for an IPv6 router. Conversely, if a static IPv6 route fails or if you want to change your network topology, you might need to manually reconfigure the static IPv6 route.

Configuring a static IPv6 route

You can configure a static IPv6 route to be redistributed into a routing protocol, but you cannot redistribute routes learned by a routing protocol into the static IPv6 routing table.

Before configuring a static IPv6 route, you must enable the forwarding of IPv6 traffic on the router using the **ipv6 unicast-routing** command and enable IPv6 on at least one interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface. For more information on performing these configuration tasks, refer to 38, “Configuring Basic IPv6 Connectivity”.

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32, a next-hop gateway with the global address 4fee:2343:0:ee44::1, and an administrative distance of 110, enter the following command.

```
NetIron(config)# ipv6 route 8eff::0/32 4fee:2343:0:ee44::1 distance 110
```

Syntax: **[no] ipv6 route** <*dest-ipv6-prefix*>/<*prefix-length*> <*next-hop-ipv6-address*> [<*metric*>] [*distance* <*number*>]

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32 and a next-hop gateway with the link-local address fe80::1 that the router can access through Ethernet interface 3/1, enter the following command.

```
NetIron(config)# ipv6 route 8eff::0/32 ethernet 1 fe80::1
```

Syntax: **[no] ipv6 route** <*dest-ipv6-prefix*>/<*prefix-length*> [**ethernet** <*slot/port*> | **pos** <*slot/port*> | **ve** <*num*> | **null0**] <*next-hop-ipv6-address*> [<*metric*>] [**tag** <num>] [**distance** <*number*>]

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32 and a next-hop gateway that the router can access through tunnel 1, enter the following command.

```
NetIron(config)# ipv6 route 8eff::0/32 tunnel 1
```

Syntax: **[no] ipv6 route** <*dest-ipv6-prefix*>/<*prefix-length*> <*interface*> <*port*> [<*metric*>] [**distance** <*number*>]

Table 313 describes the parameters associated with this command and indicates the status of each parameter.

TABLE 313 Static IPv6 route parameters

Parameter	Configuration details	Status
The IPv6 prefix and prefix length of the route's destination network.	You must specify the <*dest-ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.	Mandatory for all static IPv6 routes.
The route's next-hop gateway, which can be one of the following: • The IPv6 address of a next-hop gateway. • A tunnel interface.	You can specify the next-hop gateway as one of the following types of IPv6 addresses: • A global address. • A link-local address. If you specify a global address, you do not need to specify any additional parameters for the next-hop gateway. If you specify a link-local address, you must also specify the interface through which to access the address. You can specify one of the following interfaces: • An Ethernet interface. • A tunnel interface. • A virtual interface (VE). If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number. You can also specify the next-hop gateway as a tunnel interface. If you specify a tunnel interface, also specify the tunnel number.	Mandatory for all static IPv6 routes.
The route's metric.	You can specify a value from 1 – 16.	Optional for all static IPv6 routes. (The default metric is 1.)
Tag <*number*>	This parameter specifies the tag value of the route.	The possible values are 0 - 4294967295. The default is 0.
The route's administrative distance.	You must specify the **distance** keyword and any numerical value.	Optional for all static IPv6 routes. (The default administrative distance is 1.)

A metric is a value that the router uses when comparing this route to other static routes in the IPv6 static route table that have the same destination. The metric applies only to routes that the router has already placed in the IPv6 static route table.

The administrative distance is a value that the router uses to compare this route with routes from other route sources that have the same destination. (The router performs this comparison before placing a route in the IPv6 route table.) This parameter does not apply to routes that are already in the IPv6 route table. In general, a low administrative distance indicates a preferred route. By default, static routes take precedence over routes learned by routing protocols. If you want a dynamic route to be chosen over a static route, you can configure the static route with a higher administrative distance than the dynamic route.

Configuring a IPv6 static multicast route

IPv6 multicast routes allow you to control the network path used by multicast traffic. Static multicast routes are especially useful when the unicast and multicast topologies of a network are different. You can avoid the need to make the topologies similar by instead configuring static multicast routes.

NOTE
This feature is not supported for DVMRP.

You can configure more than one static IPv6 multicast route. The PowerConnect by default uses the most specific route that matches a multicast source address. You can also specify route preference using the **route-preference** command as described in "Route selection precedence for multicast" on page 1170. Thus, if you want to configure a multicast static route for a specific multicast source and also configure another multicast static route for all other sources, you can configure two static routes.

To configure a IPv6 mroute for a destination network with the prefix 8eff::0/32, a next-hop gateway with the global address 4fee:2343:0:ee44::1, and an administrative distance of 110, enter the following command.

```
NetIron(config)# ipv6 mroute 8eff::0/32 4fee:2343:0:ee44::1 distance 110
```

Syntax: [no] **ipv6 mroute** <*dest-ipv6-prefix*>/<*prefix-length*> <*next-hop-ipv6-address*> <*next-hop-enable-default*> <*next-hop-recursion*> [<*metric*>] [**distance** <*number*>] [**tag**<*number*>]

Syntax: [no] **ipv6 mroute** <*ipv6-addr*> **interface ethernet** <*slot*>/<*portnum*> | **ve** <*num*> | **tunnel** <*num*> [**distance** <*num*>] [**tag**<*number*>]

The <*ipv6-addr*> command specifies the next-hop IP address.

NOTE
In IPv6 multicasting, a route is handled in terms of its source, rather than its destination.

You can use the **ethernet** <*slot*>/<*portnum*> parameter to specify a physical port or the **ve** <*num*> parameter to specify a virtual interface.

NOTE
The **ethernet** <*slot*>/<*portnum*> parameter does not apply to PIM SM.

The **next-hop-enable-default** parameter sets the default route to resolve the static route nexthop.

The **next-hop-recursion** parameter sets the static route to resolve the static route nexthop.

The **distance** <*num*> parameter sets the administrative distance for the route. When comparing multiple paths for a route, the router prefers the path with the lower administrative distance.

NOTE
Regardless of the administrative distances, the switch always prefers directly connected routes over other routes.

Configuring RIPng 42

PowerConnect B-MLXe supports the following RIPng features:

- RIPng
- RIPng Timers
- Default Route Learning and Advertising
- Redistributing Routes Into RIPng
- Controlling Distribution of Routes through RIPng
- Distribution of Routes through RIPng
- Poison Reverse Parameters

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a distance vector (a number representing a distance) to measure the cost of a given route. RIP uses a hop count as its cost or metric.

IPv6 RIP, known as ***Routing Information Protocol Next Generation*** or ***RIPng***, functions similarly to IPv4 RIP version 2. RIPng supports IPv6 addresses and prefixes.

In addition, some new commands that are specific to RIPng have been implemented. This chapter describes the commands that are specific to RIPng. This section does not describe commands that apply to both IPv4 RIP and RIPng. For more information about these commands, refer to "Configuring RIP" on page 843.

RIPng maintains a ***Routing Information Database (RIB)***, which is a local route table. The local RIB contains the lowest-cost IPv6 routes learned from other RIP routers. In turn, RIPng attempts to add routes from its local RIB into the main IPv6 route table.

This chapter describes the following:

- How to configure RIPng.
- How to clear RIPng information from the RIPng route table.
- How to display RIPng information and statistics.

Configuring RIPng

To configure RIPng, you must do the following:

- Enable RIPng globally on the device and on individual router interfaces.

The following configuration tasks are optional:

- Change the default settings of RIPng timers.
- Configure how the device learns and advertises routes.
- Configure which routes are redistributed into RIPng from other sources.
- Configure how the device distributes routes through RIPng.

- Configure poison reverse parameters.

Enabling RIPng

Before configuring the device to run RIPng, you must do the following:

- Enable the forwarding of IPv6 traffic on the device using the **ipv6 unicast-routing** command.
- Enable IPv6 on each interface over which you plan to enable RIPng. You enable IPv6 on an interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

For more information about performing these configuration tasks, refer to 38, "Configuring Basic IPv6 Connectivity".

By default, RIPng is disabled. To enable RIPng, you must enable it globally on the device and also on individual router interfaces.

NOTE
Enabling RIPng globally on the device does not enable it on individual router interfaces.

To enable RIPng globally, enter the following command.

```
NetIron(config-rip-router)#ipv6 router rip
NetIron(config-ripng-router)#
```

After you enter this command, the device enters the RIPng configuration level, where you can access several commands that allow you to configure RIPng.

Syntax: [no] ipv6 router rip

To disable RIPng globally, use the **no** form of this command.

After enabling RIPng globally, you must enable it on individual router interfaces. You can enable it on physical as well as virtual routing interfaces. For example, to enable RIPng on Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 rip enable
```

Syntax: [no] ipv6 rip enable

To disable RIPng on an individual router interface, use the **no** form of this command.

Configuring RIPng timers

Table 314 describes the RIPng timers and provides their defaults.

TABLE 314 RIPng timers

Timer	Description	Default
Update	Amount of time (in seconds) between RIPng routing updates.	30 seconds.
Timeout	Amount of time (in seconds) after which a route is considered unreachable.	180 seconds.
Hold-down	Amount of time (in seconds) during which information about other paths is ignored.	180 seconds.
Garbage-collection	Amount of time (in seconds) after which a route is removed from the routing table.	120 seconds.

You can adjust these timers for RIPng. Before doing so, keep the following caveats in mind:

- If you adjust these RIPng timers, it is strongly recommended to set the same timer values for all routerRouting Switches and access servers in the network.
- Setting the update timer to a shorter interval can cause the routerRouting Switches to spend excessive time updating the IPv6 route table.
- It is recommended to set the timeout timer value to at least three times the value of the update timer.
- It is recommended that a shorter hold-down timer interval, because a longer interval can cause delays in RIPng convergence.

The following example sets updates to be broadcast every 45 seconds. If a route is not heard from in 135 seconds, the route is declared unusable. Further information is suppressed for an additional 10 seconds. Assuming no updates, the route is flushed from the routing table 20 seconds after the end of the hold-down period.

```
NetIron(config)# ipv6 router rip
NetIron(config-ripng-router)# timers 45 135 10 20
```

Syntax: **[no] timers** <*update-timer*> <*timeout-timer*> <*hold-down-timer*> <*garbage-collection-timer*>

Possible values for the timers are as follows:

- **Update timer:** 3 – 65535 seconds.
- **Timeout timer:** 9 – 65535 seconds.
- **Hold-down timer:** 9 – 65535 seconds.
- **Garbage-collection timer:** 9 – 65535 seconds.

NOTE
You must enter a value for each timer, even if you want to retain the current setting of a particular timer.

To return to the default values of the RIPng timers, use the **no** form of this command.

Configuring route learning and advertising parameters

You can configure the following learning and advertising parameters:

- Learning and advertising of RIPng default routes.
- Advertising of IPv6 address summaries.
- Metric of routes learned and advertised on a router interface.

Configuring default route learning and advertising

By default, the device does not learn IPv6 default routes (::/0). You can originate default routes into RIPng, which causes individual router interfaces to include the default routes in their updates. When configuring the origination of the default routes, you can also do the following:

- Suppress all other routes from the updates.
- Include all other routes in the updates.

For example, to originate default routes in RIPng and suppress all other routes in updates sent from Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 rip default-information only
```

To originate IPv6 default routes and include all other routes in updates sent from Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 rip default-information originate
```

Syntax: [no] ipv6 rip default-information only | originate

The **only** keyword originates the default routes and suppresses all other routes from the updates.

The **originate** keyword originates the default routes and includes all other routes in the updates.

To remove the explicit default routes from RIPng and suppress advertisement of these routes, use the **no** form of this command.

Advertising IPv6 address summaries

You can configure RIPng to advertise a summary of IPv6 addresses from a router interface and to specify an IPv6 prefix that summarizes the routes.

If a route's prefix length matches the value specified in the **ipv6 rip summary-address** command, RIPng advertises the prefix specified in the **ipv6 rip summary-address** command instead of the original route.

For example, to advertise the summarized prefix 2001:469e::/36 instead of the IPv6 address 2001:469e:0:adff:8935:e838:78:e0ff with a prefix length of 64 bits from Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 address 2001:469e:0:adff:8935:e838:78:
e0ff /64
NetIron(config-if-e100-3/1)# ipv6 rip summary-address 2001:469e::/36
```

Syntax: [no] ipv6 rip summary-address <*ipv6-prefix*>/<*prefix-length*>

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

To stop the advertising of the summarized IPv6 prefix, use the **no** form of this command.

Changing the metric of routes learned and advertised on an interface

A router interface increases the metric of an incoming RIPng route it learns by an offset (the default is one). The device then places the route in the route table. When the device sends an update, it advertises the route with the metric plus the default offset of zero in an outgoing update message.

You can change the metric offset an individual interface adds to a route learned by the interface or advertised by the interface. For example, to change the metric offset for incoming routes learned by Ethernet interface 3/1 to one and the metric offset for outgoing routes advertised by the interface to three, enter the following commands.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 rip metric-offset 1
NetIron(config-if-e100-3/1)# ipv6 rip metric-offset out 3
```

In this example, if Ethernet interface 3/1 learns about an incoming route, it will increase the incoming metric by two (the default offset of 1 and the additional offset of 1 as specified in this example). If Ethernet interface 3/1 advertises an outgoing route, it will increase the metric by 3 as specified in this example.

Syntax: [no] ipv6 rip metric-offset [out] <*1 – 16*>

To return the metric offset to its default value, use the **no** form of this command.

Redistributing routes into RIPng

You can configure the device to redistribute routes from the following sources into RIPng:

- IPv6 static routes.
- Directly connected IPv6 networks.
- BGP4+.
- IPv6 IS-IS.
- OSPFv3.

When you redistribute a route from BGP4+, IPv6 IS-IS, or OSPFv3 into RIPng, the device can use RIPng to advertise the route to its RIPng neighbors.

When configuring the device to redistribute routes, such as BGP4+ routes, you can optionally specify a metric for the redistributed routes. If you do not explicitly configure a metric, the default metric value of one is used.

For example, to redistribute OSPFv3 routes into RIPng, enter the following command.

```
NetIron(config)# ipv6 router rip
NetIron(config-ripng-router)# redistribute ospf
```

Syntax: [no] redistribute bgp | connected | isis | ospf | static [metric <*number*>**]**

For the metric, specify a numerical value that is consistent with RIPng.

Controlling distribution of routes through RIPng

You can create a prefix list and then apply it to RIPng routing updates that are received or sent on a router interface. Performing this task allows you to control the distribution of routes through RIPng.

For example, to permit the inclusion of routes with the prefix 2001::/16 in RIPng routing updates sent from Ethernet interface 3/1, enter the following commands.

```
NetIron(config)# ipv6 prefix-list routesfor2001 permit 2001::/16
NetIron(config)# ipv6 router rip
NetIron(config-ripng-router)# distribute-list prefix-list routesfor2001 out
ethernet 3/1
```

To deny prefix lengths greater than 64 bits in routes that have the prefix 3EE0:A99::/64 and allow all other routes received on tunnel interface 3/1, enter the following commands.

```
NetIron(config)# ipv6 prefix-list 3ee0routes deny 3ee0:a99::/64 le 128
NetIron(config)# ipv6 prefix-list 3ee0routes permit ::/0 ge 0 le 128
NetIron(config)# ipv6 router rip
NetIron(config-ripng-router)# distribute-list prefix-list 3ee0routes in
tunnel 1
```

For information about prefix lists, including the syntax of the **ipv6 prefix-list** command, refer to Chapter 39, "Configuring an IPv6 Prefix List".

Syntax: [no] distribute-list prefix-list <*name*> **in | out** <*interface*> <*port*>

The <*name*> parameter indicates the name of the prefix list generated using the **ipv6 prefix-list** command.

The **in** keyword indicates that the prefix list is applied to incoming routing updates on the specified interface.

The **out** keyword indicates that the prefix list is applied to outgoing routing updates on the specified interface.

For the <*interface*> parameter, you can specify the **ethernet**, **pos**, **loopback**, **ve**, or **tunnel** keywords. If you specify an Ethernet or POS interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number.

To remove the distribution list, use the **no** form of this command.

Configuring poison reverse parameters

By default, poison reverse is disabled on a RIPng router. If poison reverse is enabled, RIPng advertises routes it learns from a particular interface over that same interface with a metric of 16, which means that the route is unreachable.

If poison reverse is enabled on the RIPng router, it takes precedence over split horizon (if it is also enabled).

To enable poison reverse on the RIPng router, enter the following commands.

```
NetIron(config)# ipv6 router rip
NetIron(config-ripng-router)# poison-reverse
```

Syntax: [no] poison-reverse

To disable poison-reverse, use the **no** version of this command.

By default, if a RIPng interface goes down, the device does not send a triggered update for the interface's IPv6 networks.

To better handle this situation, you can configure a RIPng router to send a triggered update containing the local routes of the disabled interface with an unreachable metric of 16 to the other RIPng routers in the routing domain. You can enable the sending of a triggered update by entering the following commands.

```
NetIron(config)# ipv6 router rip
NetIron(config-ripng-router)# poison-local-routes
```

Syntax: [no] poison-local-routes

To disable the sending of a triggered update, use the **no** version of this command.

Clearing RIPng routes from IPv6 route table

To clear all RIPng routes from the RIPng route table and the IPv6 main route table and reset the routes, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 rip routes
```

Syntax: clear ipv6 rip routes

Displaying RIPng information

You can display the following RIPng information:

- RIPng configuration.
- RIPng routing table.

Displaying RIPng configuration

To display RIPng configuration information, enter the following command at any CLI level.

```
NetIron# show ipv6 rip
 IPv6 rip enabled, port 521
     Administrative distance is 120
     Updates every 30 seconds, expire after 180
     Holddown lasts 180 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
     Periodic updates 0, trigger updates 0
     Distribute List, Inbound : Not set
     Distribute List, Outbound : Not set
     Redistribute: CONNECTED
```

Syntax: show ipv6 rip

This display shows the following information.

TABLE 315 RIPng configuration fields

This field...	Displays...
IPv6 RIP status or port	The status of RIPng on the device. Possible status is “enabled” or “disabled.” The UDP port number over which RIPng is enabled.
Administrative distance	The setting of the administrative distance for RIPng.
Updates or expiration	The settings of the RIPng update and timeout timers.
Holddown or garbage collection	The settings of the RIPng hold-down and garbage-collection timers.
Split horizon or poison reverse	The status of the RIPng split horizon and poison reverse features. Possible status is “on” or “off.”
Default routes	The status of RIPng default routes.
Periodic updates or trigger updates	The number of periodic updates and triggered updates sent by the RIPng router.

TABLE 315 RIPng configuration fields (Continued)

This field...	Displays...
Distribution lists	The inbound and outbound distribution lists applied to RIPng.
Redistribution	The types of IPv6 routes redistributed into RIPng. The types can include the following: • STATIC – IPv6 static routes are redistributed into RIPng. • CONNECTED – Directly connected IPv6 networks are redistributed into RIPng. • BGP – BGP4+ routes are redistributed into RIPng. • ISIS – IPv6 IS-IS routes are redistributed into RIPng. • OSPF – OSPFv3 routes are redistributed into RIPng.

Displaying RIPng routing table

To display the RIPng routing table, enter the following command at any CLI level.

```
NetIron# show ipv6 rip route
IPv6 RIP Routing Table - 4 entries:
 2000:4::/64, from ::, null   (0)
          CONNECTED, metric 1, tag 0, timers: none
 2002:c0a8:46a::/64, from ::, null   (1)
          CONNECTED, metric 1, tag 0, timers: none
 2999::1/128, from ::, null   (2)
          CONNECTED, metric 1, tag 0, timers: none
 5000:2::/64, from ::, null   (3)
          CONNECTED, metric 1, tag 0, timers: none
```

Syntax: show ipv6 rip route [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*>]

The <*ipv6-prefix*>/<*prefix-length*> parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The <*ipv6-address*> parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

This display shows the following information.

TABLE 316 RIPng routing table fields

This field...	Displays...
RIPng Routing Table entries	The total number of entries in the RIPng routing table.
<*ipv6-prefix*>/<*prefix-length*> <*ipv6-address*>	The IPv6 prefix and prefix length. The IPv6 address.
Next-hop router	The next-hop router for this device. If :: appears, the route is originated locally.
Interface	The interface name. If “null” appears, the interface is originated locally.

TABLE 316 RIPng routing table fields

This field...	Displays...
Source of route	The source of the route information. The source can be one of the following: • RIP – routes learned by RIPng. • CONNECTED – IPv6 routes redistributed from directly connected networks. • STATIC – IPv6 static routes are redistributed into RIPng. • BGP – BGP4+ routes are redistributed into RIPng. • ISIS – IPv6 IS-IS routes are redistributed into RIPng. • OSPF – OSPFv3 routes are redistributed into RIPng.
Metric <*number*>	The cost of the route. The <*number*> parameter indicates the number of hops to the destination.
Tag <*number*>	The tag value of the route.
Timers:	Indicates if the hold-down timer or the garbage-collection timer is set.

Chapter

Configuring OSPF Version 3

43

The following OSPF Version 3 features supported by PowerConnect B-MLXe Series.

- OSPF Version 3
- Link-State Advertisement Router LSAs (Type 1)
- Link-State Advertisement Network LSAs (Type 2)
- Link-State Advertisement Interarea-prefix LSAs for ABRs (Type 3)
- Link-State Advertisement Interarea-router LSAs for ASBRs (Type 4)
- Link-State Advertisement Autonomous system external LSAs (Type 5)
- Link-State Advertisement Link LSAs (Type 8)
- Link-State Advertisement
- Intra-area prefix LSAs (Type 9)
- IPsec for OSPFv3
- New encryption code for passwords, authentication keys, and community strings
- Redistributing Routes into OSPFv3
- Filtering OSPFv3 Routes
- Default Route Origination
- Shortest Path First Timers
- Event Logging

Open Shortest Path First (OSPF) is a link-state routing protocol. OSPF uses link-state advertisements (LSAs) to update neighboring routers about its interfaces and information on those interfaces. The router floods LSAs to all neighboring routers to update them about the interfaces. Each router maintains an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router.

OSPF Version 3

IPv6 supports OSPF Version 3 (OSPFv3), which functions similarly to OSPF Version 2 (OSPFv2), the current version that IPv4 supports, except for the following enhancements:

- Support for IPv6 addresses and prefixes.
- Ability to configure several IPv6 addresses on a router interface. (OSPFv3 imports all or none of the the address prefixes configured on a router interface. You cannot select the addresses to import.)
- Ability to run one instance of OSPF Version 2 and one instance of OSPFv3 concurrently on a link.
- IPv6 link-state advertisements (LSAs).

In addition, some new commands that are specific to OSPFv3 have been implemented. This section describes the commands that are specific to OSPFv3.

NOTE
Although OSPF Version 2 and OSPF 3 function similarly to each other, Dell has implemented the user interface for each version independently of one another. Therefore, any configuration of OSPFv2 features will not affect the configuration of OSPFv3 features and vice versa.

Link-state advertisement types for OSPFv3

OSPFv3 supports the following types of LSAs:

- Router LSAs (Type 1)
- Network LSAs (Type 2)
- Interarea-prefix LSAs for ABRs (Type 3)
- Interarea-router LSAs for ASBRs (Type 4)
- Autonomous system external LSAs (Type 5)
- Link LSAs (Type 8)
- Intra-area-prefix LSAs (Type 9)

For more information about these LSAs, refer to RFC 2740.

Configuring OSPFv3

To configure OSPFv3, you must perform the following steps.

- Enable OSPFv3 globally.
- Assign OSPF areas.
- Assign router interfaces to an OSPF area.

The following configuration tasks are optional:

- Configure a virtual link between an Area Border Router (ABR) without a physical connection to a backbone area and the device in the same area with a physical connection to the backbone area.
- Change the reference bandwidth for the cost on OSPFv3 interfaces.
- Configure the redistribution of routes into OSPFv3.
- Configure default route origination.
- Modify the shortest path first (SPF) timers.
- Modify the administrative distances for OSPFv3 routes.
- Configure the OSPFv3 LSA pacing interval.
- Modify how often the device checks on the elimination of the database overflow condition.
- Modify the external link state database limit.
- Modify the default values of OSPFv3 parameters for router interfaces.
- Disable or re-enable OSPFv3 event logging.

Enabling OSPFv3

Before enabling the device to run OSPFv3, you must perform the following steps.

- Enable the forwarding of IPv6 traffic on the device using the **ipv6 unicast-routing** command.
- Enable IPv6 on each interface over which you plan to enable OSPFv3. You enable IPv6 on an interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

For more information about performing these configuration tasks, refer to 38, "Configuring Basic IPv6 Connectivity".

By default, OSPFv3 is disabled. To enable OSPFv3 for a default Virtual Routing and Forwarding (VRF), you must enable it globally.

To enable OSPFv3 globally, enter the following command.

```
NetIron(config)# ipv6 router ospf
NetIron(config-ospf6-router)#
```

After you enter this command, the device enters the IPv6 OSPF configuration level, where you can access several commands that allow you to configure OSPFv3.

Syntax: [no] ipv6 router ospf [vrf]

To disable OSPFv3, enter the **no** form of this command. If you disable OSPFv3, the device removes all the configuration information for the disabled protocol from the running-configuration file. Moreover, when you save the configuration to the startup-config file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup-config file.

When you disable OSPFv3, the following warning message is displayed on the console.

```
NetIron(config-ospf6-router)# no ipv6 router ospf
ipv6 router ospf mode now disabled. All ospf config data will be lost when writing
to flash!
```

If you have disabled the protocol but have not yet saved the configuration to the startup-config file and reloaded the software, you can restore the configuration information by re-entering the command to enable the protocol (for example, **ipv6 router ospf**). If you have already saved the configuration to the startup-config file and reloaded the software, the configuration information is gone. If you are testing an OSPF configuration and are likely to disable and re-enable the protocol, you should make a backup copy of the startup-config file containing the protocol configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

Assigning OSPFv3 areas

After OSPFv3 is enabled, you can assign OSPFv3 areas. You can assign an IPv4 address or a number as the *area ID* for each area. The area ID is representative of all IPv6 addresses (subnets) on a router interface. Each router interface can support one area.

An area can be *normal* or a *stub*:

- Normal – OSPF routers within a normal area can send and receive external Link State Advertisements (LSAs).

- Stub – OSPF routers within a stub area cannot send or receive external LSAs. In addition, OSPF routers in a stub area must use a default route to the Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) of the area to send traffic out of the area.

For example, to set up OSPFv3 areas 0.0.0.0, 200.5.0.0, 192.5.1.0, and 195.5.0.0, enter the following commands.

```
NetIron(config-ospf6-router)# area 0.0.0.0
NetIron(config-ospf6-router)# area 200.5.0.0
NetIron(config-ospf6-router)# area 192.5.1.0
NetIron(config-ospf6-router)# area 195.5.0.0
```

Syntax: [no] area <*number*> **|** <*ipv4-address*>

The <*number*> | <*ipv4-address*> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 through 2,147,483,647.

NOTE
You can assign only one area on a router interface.

Assigning a totally stubby area

By default, the device sends summary LSAs (type 3 LSAs) into stub areas. You can reduce the number of LSAs sent into a stub area by configuring the device to stop sending summary LSAs into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs into a stub area, but the device still accepts summary LSAs from OSPF neighbors and floods them to other areas. The device can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each router.

When you disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the router flushes all of the summary LSAs it has generated (as an ABR) from the area.

NOTE
This feature applies only when the device is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

For example, to disable summary LSAs for stub area 40 and specify an additional metric of 99, enter the following command.

```
NetIron(config-ospf6-router)# area 40 stub 99 no-summary
```

Syntax: [no] area <*number*> **|** <*ipv4-address*> **stub** <*metric*> **[no-summary]**

The <*number*> | <*ipv4-address*> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 through 2,147,483,647.

The **stub** <*metric*> parameter specifies an additional cost for using a route to or from this area and can be from 1 through 16777215. There is no default. Normal areas do not use the cost parameter.

The **no-summary** parameter applies only to stub areas and disables summary LSAs from being sent into the area.

Assigning interfaces to an area

After you define OSPFv3 areas, you must assign router interfaces to the areas. All router interfaces must be assigned to one of the defined areas on an OSPF router. When an interface is assigned to an area, all corresponding subnets on that interface are automatically included in the assignment.

For example, to assign Ethernet interface 3/1 to area 192.5.0.0, enter the following commands.

```
NetIron(config)# interface Ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 ospf area 195.5.0.0
```

Syntax: [no] ipv6 ospf area <*number*> **|** <*ipv4-address*>

The <*number*> | <*ipv4-address*> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 through 2,147,483,647.

To remove the interface from the specified area, use the **no** form of this command.

Specifying a network type

You can specify a point-to-point or broadcast network type for any OSPF interface of the following types: POS, Ethernet, or VE interface. To specify the network type for an OSPF interface, use the following commands.

```
NetIron(config)# interface pos 3/1
NetIron(config-if-e100-3/1)# ipv6 ospf network broadcast
```

Syntax: [no] ipv6 ospf network point-to-point | broadcast

The **point-to-point** parameter specifies that the OSPF interface will support point-to-point networking. This is the default setting for POS interfaces.

The **broadcast** parameter specifies that the OSPF interface will support broadcast networking. This is the default setting for Ethernet and VE interfaces.

The **no** form of the command disables the command configuration.

Configuring virtual links

All ABRs must have either a direct or indirect link to an OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to a backbone area, you can configure a virtual link from the ABR to another router within the same area that has a physical connection to the backbone area.

The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection) and the ABR requiring a logical connection to the backbone.

Two parameters must be defined for all virtual links—transit area ID and neighbor router:

- The transit area ID represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.
- The neighbor router is the router ID (IPv4 address) of the router that is physically connected to the backbone when assigned from the router interface requiring a logical connection. The neighbor router is the router ID (IPv4 address) of the router requiring a logical connection to the backbone when assigned from the router interface with the physical connection.

NOTE
By default, the router ID is the IPv4 address configured on the lowest-numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest-numbered IPv4 address configured on the device.

When you establish an area virtual link, you must configure it on both ends of the virtual link. For example, imagine that ABR1 in areas 1 and 2 is cut off from the backbone area (area 0). To provide backbone access to ABR1, you can add a virtual link between ABR1 and ABR2 in area 1 using area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To define the virtual link on ABR1, enter the following command on ABR1.

```
NetIron(config-ospf6-router)# area 1 virtual-link 209.157.22.1
```

To define the virtual link on ABR2, enter the following command on ABR2.

```
NetIron(config-ospf6-router)# area 1 virtual-link 10.0.0.1
```

Syntax: **area** <*number*> **|** <*ipv4-address*> **virtual-link** <*router-id*>

The <*number*>|<*ipv4-address*> parameter specifies the transit area ID, area number, which can be a number, or in IPv4 address format. If you specify a number, the number can be from 0 through 2,147,483,647.

The <*router-id*> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a router, enter the **show ip** command.

Assigning a virtual link source address

When routers at both ends of a virtual link communicate with one another, the source address included in the packets must be a global IPv6 address. The Multi-Service IronWare software automatically selects a global IPv6 address for each transit area and advertises this address into the transit area of the Intra-area-prefix LSA. The automatically selected global IPv6 address for a transit area is the first global IPv6 address of any loopback interface in the transit area. If no global IPv6 address is available on a loopback interface in the area, then the first global IPv6 address of the lowest-numbered interface in the UP state (belonging to the transit area) will be assigned. If no global IPv6 address is configured on any of the OSPF interfaces in the transit area, then the virtual links in the transit area will not operate. The automatically selected IPv6 global address is updated whenever the previously selected IPv6 address of the interface changes, is removed, or if the interface goes down.

NOTE
The existing selected virtual link address will not change because the global IPv6 address is now available on a loopback interface or a lower-numbered interface in the transit area. To force the global IPv6 address for the virtual link to be the global IPv6 address of a newly configured loopback, or a lower-numbered interface in the area, you will have to either disable the existing selected interface or remove the currently selected global IPv6 address from the interface.

Modifying virtual link parameters

You can modify the following virtual link parameters:

- **Dead-interval:** The number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router is down. The range is from 1 through 65535 seconds. The default is 40 seconds.
- **Hello-interval:** The length of time between the transmission of hello packets. The range is from 1 through 65535 seconds. The default is 10 seconds.
- **Retransmit-interval:** The interval between the retransmission of link state advertisements to router adjacencies for this interface. The range is from 0 through 3600 seconds. The default is 5 seconds.
- **Transmit-delay:** The period of time it takes to transmit Link State Update packets on the interface. The range is from 0 through 3600 seconds. The default is 1 second.

NOTE
The values of the **dead-interval** and **hello-interval** parameters must be the same at both ends of a virtual link. Therefore, if you modify the values of these parameters at one end of a virtual link, you must make the same modifications on the other end of the link.

The values of the other virtual link parameters do not require synchronization.

For example, to change the **dead-interval** parameter to 60 seconds on the virtual links defined on ABR1 and ABR2, enter the following command on ABR1.

```
NetIron(config-ospf6-router)# area 1 virtual-link 209.157.22.1
dead-interval 60
```

Enter the following command on ABR2.

```
NetIron(config-ospf6-router)# area 1 virtual-link 10.0.0.1 dead-interval 60
```

Syntax: **area** <*number*> **|** <*ipv4-address*> **virtual-link** <*router-id*> **[dead-interval** <*seconds*> **| hello-interval** <*seconds*> **| retransmit-interval** <*seconds*> **| transmit-delay** <*seconds*>**]**

The **area** <*number*> | <*ipv4-address*> parameter specifies the transit area ID.

The <*router-id*> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a router, enter the **show ip** command.

The **dead-interval**, **hello-interval**, **retransmit-interval**, and **transmit-delay** parameters are described earlier in this section.

Changing the reference bandwidth for the cost on OSPFv3 interfaces

Each interface on which OSPFv3 is enabled has a cost associated with it. The device advertises its interfaces and their costs to OSPFv3 neighbors. For example, if an interface has an OSPF cost of 10, the device advertises the interface with a cost of 10 to other OSPF routers.

By default, OSPF cost of an interface is based on the port speed of the interface. The software uses the following formula to calculate the cost.

Cost = reference-bandwidth/interface-speed

By default, the reference bandwidth is 100 Mbps. If the resulting cost is less than 1, the software rounds the cost up to 1. The default reference bandwidth results in the following costs:

- 10 Mbps port cost = 100/10 = 10
- 100 Mbps port cost = 100/100 = 1

- 1000 Mbps port cost = 100/1000 = 0.10, which is rounded up to 1
- 155 Mbps port cost = 100/155 = 0.65, which is rounded up to 1
- 622 Mbps port cost = 100/622 = 0.16, which is rounded up to 1
- 2488 Mbps port cost = 100/2488 = 0.04, which is rounded up to 1

The interfaces that consist of more than one physical port is calculated as follows:

- LAG group – The combined bandwidth of all the ports.
- Virtual (Ethernet) interface – The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

You can change the default reference bandwidth from 100 Mbps to a value from 1 through 4294967 Mbps.

If a change to the reference bandwidth results in a cost change to an interface, the device sends a link-state update to update the costs of interfaces advertised by the device.

NOTE
If you specify a cost for an interface, your specified cost overrides the cost that the software calculates.

Some interface types are not affected by the reference bandwidth and always have the same cost regardless of the reference bandwidth in use:

- The cost of a loopback interface is always 1.
- The cost of a virtual link is calculated using the Shortest Path First (SPF) algorithm and is not affected by the auto-cost feature.
- The bandwidth for tunnel interfaces is 9 Kbps and is subject to the auto-cost feature.

For example, to change the reference bandwidth to 500, enter the following command.

```
NetIron(config-ospf6-router)# auto-cost reference-bandwidth 500
```

The reference bandwidth specified in this example results in the following costs:

- 10 Mbps port cost = 500/10 = 50
- 100 Mbps port cost = 500/100 = 5
- 1000 Mbps port cost = 500/1000 = 0.5, which is rounded up to 1
- 155 Mbps port cost = 500/155 = 3.23, which is rounded up to 4
- 622 Mbps port cost = 500/622 = 0.80, which is rounded up to 1
- 2488 Mbps port cost = 500/2488 = 0.20, which is rounded up to 1

The costs for 10 Mbps, 100 Mbps, and 155 Mbps ports change as a result of the changed reference bandwidth. Costs for higher-speed interfaces remain the same.

Syntax: [no] auto-cost reference-bandwidth <*number*>

The <*number*> parameter specifies the reference bandwidth in the range from 1 through 4294967. The default is 100.

To restore the reference bandwidth to its default value and thus restore the default costs of the interfaces to their default values, enter the **no** form of this command.

Redistributing routes into OSPFv3

In addition to specifying which routes are redistributed into OSPFv3, you can configure the following aspects related to route redistribution:

- Default metric.
- Metric type.
- Advertisement of an external aggregate route.

Configuring route redistribution into OSPFv3

You can configure the device to redistribute routes from the following sources into OSPFv3:

- IPv6 static routes.
- Directly connected IPv6 networks.
- BGP4+.
- IPv6 IS-IS.
- RIPng.

You can redistribute routes in the following ways:

- By route types, for example, the device redistributes all IPv6 static and RIPng routes.
- By using a route map to filter which routes to redistribute, for example, the device redistributes specified IPv6 static and RIPng routes only.

For example, to configure the redistribution of all IPv6 static, RIPng, and IPv6 IS-IS level-1 and level-2 routes, enter the following commands.

```
NetIron(config-ospf6-router)# redistribute static
NetIron(config-ospf6-router)# redistribute rip
NetIron(config-ospf6-router)# redistribute isis level-1-2
```

Syntax: [no] **redistribute bgp | connected | isis [level-1 | level-1-2 | level-2] | rip | static [metric** <*number*> **| metric-type** <*type*>]

The **bgp** | **connected** | **isis** | **rip** | **static** keywords specify the route source.

The **level-1** | **level-1-2** | **level-2** keywords (for IPv6 IS-IS only) allow you to specify that the device redistributes level-1 routes only, level-2 routes only, or both level-1 and level-2 routes.

The **metric** <*number*> parameter specifies the metric used for the redistributed route. If a value is not specified for this option, and the value for the **default-metric** command is set to 0, its default metric, then routes redistributed from the various routing protocols will have the metric value of the protocol from which they are redistributed. For information about the **default-metric** command, refer to "Modifying default metric for routes redistributed into OSPF Version 3" on page 1817.

The **metric-type** <*type*> parameter specifies an OSPF metric type for the redistributed route. You can specify external type 1 or external type 2. If a value is not specified for this option, the device uses the value specified by the **metric-type** command. For information about modifying the default metric type using the **metric-type** command, refer to "Modifying metric type for routes redistributed into OSPF Version 3" on page 1817.

For example, to configure a route map and use it for redistribution of routes into OSPFv3, enter commands such as the following.

```
NetIron(config)# ipv6 route 2001:1::/32 4823:eoff:343e::23
NetIron(config)# ipv6 route 2001:2::/32 4823:eoff:343e::23
NetIron(config)# ipv6 route 2001:3::/32 4823:eoff:343e::23 metric 5
NetIron(config)# route-map abc permit 1
NetIron(config-routemap abc)# match metric 5
NetIron(config-routemap abc)# set metric 8
NetIron(config-routemap abc)# ipv6 router ospf
NetIron(config-ospf6-router)# redistribute static route-map abc
```

The commands in this example configure some static IPv6 routes and a route map, and use the route map for redistributing the static IPv6 routes into OSPFv3.

The **ipv6 route** commands configure the static IPv6 routes. The route-map command begins configuration of a route map called “abc”. The number indicates the route map entry (called the “instance”) you are configuring. A route map can contain multiple entries. The software compares packets to the route map entries in ascending numerical order and stops the comparison once a match is found.

The **match** command in the route map matches on routes that have 5 for their metric value (cost). The **set** command changes the metric in routes that match the route map to 8.

The **redistribute** command configures the redistribution of static IPv6 routes into OSPFv3, and uses route map “abc“ to control the routes that are redistributed. In this example, the route map allows a static IPv6 route to be redistributed into OSPF only if the route has a metric of 5, and changes the metric to 8 before placing the route into the OSPF route redistribution table.

Syntax: [no] redistribute bgp | connected | isis | rip | static [route-map <*map-name*>**]**

The **bgp** | **connected** | **isis** | **rip** | **static** keywords specify the route source.

The **route-map** <*map-name*> parameter specifies the route map name. The following match parameters are valid for OSPFv3 redistribution:

- **match ip address** | **next-hop** <*acl-number*>
- **match metric** <*number*>
- **match tag** <*tag-value*>

The following set parameters are valid for OSPF redistribution:

- **set ip next hop** <*ipv4-address*>
- **set metric** [+ | -] <*number*> | none
- **set metric-type** type-1 | type-2
- **set tag** <*tag-value*>

NOTE
You must configure the route map before you configure a redistribution filter that uses the route map.

NOTE
When you use a route map for route redistribution, the software disregards the permit or deny action of the route map.

NOTE
For an external route that is redistributed into OSPFv3 through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map or the **default-metric** <*num*> command. For a route redistributed without using a route map, the metric is set by the metric parameter if set or the **default-metric** <*num*> command if the metric parameter is not set.

Modifying default metric for routes redistributed into OSPF Version 3

The default metric is a global parameter that specifies the cost applied by default to routes redistributed into OSPFv3. The default value is 10.

If the **metric** parameter for the **redistribute** command is not set and the **default-metric** command is set to 10, its default value, then routes redistributed from the various routing protocols will have the metric value of the protocol from which they are redistributed. For information about the **redistribute** command, refer to "Configuring route redistribution into OSPFv3" on page 1815.

NOTE
You also can define the cost on individual interfaces. The interface cost overrides the default cost. For information about defining the cost on individual interfaces, refer to "Modifying OSPFv3 interface defaults" on page 1825 and "Changing the reference bandwidth for the cost on OSPFv3 interfaces" on page 1813.

To assign a default metric of 4 to all routes imported into OSPFv3, enter the following command.

```
NetIron(config-ospf6-router)# default-metric 4
```

Syntax: [no] default-metric <*number*>

You can specify a value from 0 – 65535. The default is 0.

To restore the default metric to the default value, use the **no** form of this command.

Modifying metric type for routes redistributed into OSPF Version 3

The device uses the **metric-type** parameter by default for all routes redistributed into OSPFv3 unless you specify a different metric type for individual routes using the **redistribute** command. (For more information about using the **redistribute** command, refer to "Redistributing routes into OSPFv3" on page 1815.)

A type 1 route specifies a small metric (two bytes), while a type 2 route specifies a big metric (three bytes). The default value is type 2.

To modify the default value of type 2 to type 1, enter the following command.

```
NetIron(config-ospf6-router)# metric-type type1
```

Syntax: [no] metric-type type1 | type2

To restore the metric type to the default value, use the **no** form of this command.

Configuring external route summarization

When the device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to advertise one external route as an aggregate for all redistributed routes that are covered by a specified IPv6 address range.

When you configure an address range, the range takes effect immediately. All the imported routes are summarized according to the configured address range. Imported routes that have already been advertised and that fall within the range are flushed out of the AS and a single route corresponding to the range is advertised.

If a route that falls within a configured address range is imported by the device, no action is taken if the device has already advertised the aggregate route; otherwise, the device advertises the aggregate route. If an imported route that falls within a configured address range is removed by the device, no action is taken if there are other imported routes that fall with in the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The device sets the forwarding address of the aggregate route to zero and sets the tag to zero.

If you delete an address range, the advertised aggregate route is flushed and all imported routes that fall within the range are advertised individually.

If an external link state database overflow (LSDB) condition occurs, all aggregate routes are flushed out of the AS, along with other external routes. When the device exits the external LSDB overflow condition, all the imported routes are summarized according to the configured address ranges.

NOTE
If you use redistribution filters in addition to address ranges, the device applies the redistribution filters to routes first, then applies them to the address ranges.

NOTE
If you disable redistribution, all the aggregate routes are flushed, along with other imported routes.

NOTE
This option affects only imported, type 5 external routes. A single type 5 LSA is generated and flooded throughout the AS for multiple external routes.

To configure the summary address 2201::/24 for routes redistributed into OSPFv3, enter the following command.

```
NetIron(config-ospf6-router)# summary-address 2201::/24
```

In this example, the summary prefix 2201::/24 includes addresses 2201::/1 through 2201::/24. Only the address FEC0::/24 is advertised in an external link-state advertisement.

Syntax: **summary-address** <*ipv6-prefix*>/<*prefix-length*>

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

Filtering OSPFv3 routes

You can filter the routes to be placed in the OSPFv3 route table by configuring distribution lists. OSPFv3 distribution lists can be applied globally or to an interface.

The functionality of OSPFv3 distribution lists is similar to that of OSPFv2 distribution lists. However, unlike OSPFv2 distribution lists, which filter routes based on criteria specified in an Access Control List (ACL), OSPFv3 distribution lists can filter routes using information specified in an IPv6 prefix list or a route map.

Configuration examples

The following sections show examples of filtering OSPFv3 routes using prefix lists globally and for a specific interface, as well as filtering OSPFv3 routes using a route map.

You can configure the device to use all three types of filtering. When you do this, filtering using route maps has higher priority over filtering using global prefix lists. Filtering using prefix lists for a specific interface has lower priority than the other two filtering methods.

The example in this section assume the following routes are in the OSPFv3 route table.

```
NetIron# show ipv6 ospf route

  Current Route count: 5
    Intra: 3 Inter: 0 External: 2 (Type1 0/Type2 2)
    Equal-cost multi-path: 0
    Destination                  Options   Area            Cost Type2 Cost
    Next Hop Router              Outgoing Interface
*IA 3001::/64                    --------- 0.0.0.1           0 0
    ::                           ve 10
*E2 3010::/64                    --------- 0.0.0.0          10 0
    fe80::2e0:52ff:fe00:10       ve 10
*IA 3015::/64                    V6E---R-- 0.0.0.0          11 0
    fe80::2e0:52ff:fe00:10       ve 10
*IA 3020::/64                    --------- 0.0.0.0          10 0
    ::                           ve 11
*E2 6001:5000::/64               --------- 0.0.0.0          10 0
    fe80::2e0:52ff:fe00:10       ve 10
```

Configuring an OSPFv3 distribution list using an IPv6 prefix list as input

The following example illustrates how to use an IPv6 prefix list is used to filter OSPFv3 routes.

To specify an IPv6 prefix list called filterOspfRoutes that denies route 3010::/64, enter the following commands.

```
NetIron(config)# ipv6 prefix-list filterOspfRoutes seq 5 deny 3010::/64
NetIron(config)# ipv6 prefix-list filterOspfRoutes seq 7 permit ::/0 ge 1 le 128
```

Syntax: **ipv6 prefix-list** <*name*> [**seq** <*seq-value*>] [**description** <*string*>] **deny | permit** <*ipv6-addr*>/<*mask-bits*> [**ge** <*ge-value*>] [**le** <*le-value*>]

To configure a distribution list that applies the filterOspfRoutes prefix list globally.

```
NetIron(config)# ipv6 router ospf
NetIron(config-ospf6-router)# distribute-list prefix-list filterOspfRoutes in
```

Syntax: [no] **distribute-list prefix-list** <*name*> **in** [**< ethernet** <*slot/port*> **| pos** <*slot/port*> **| ve** <*num*> **| loopback** <*num*> **]**

After this distribution list is configured, route 3010::/64 would be omitted from the OSPFv3 route table.

```
NetIron# show ipv6 ospf route

  Current Route count: 4
    Intra: 3 Inter: 0 External: 1 (Type1 0/Type2 1)
    Equal-cost multi-path: 0
    Destination                    Options   Area             Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 3001::/64                      --------- 0.0.0.1             0 0
    ::                             ve 10
*IA 3015::/64                      V6E---R-- 0.0.0.0            11 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 3020::/64                      --------- 0.0.0.0            10 0
    ::                             ve 11
*E2 6001:5000::/64                 --------- 0.0.0.0            10 0
    fe80::2e0:52ff:fe00:10         ve 10
```

The following commands specify an IPv6 prefix list called filterOspfRoutesVe that denies route 3015::/64.

```
NetIron(config)# ipv6 prefix-list filterOspfRoutesVe seq 5 deny 3015::/64
NetIron(config)# ipv6 prefix-list filterOspfRoutesVe seq 10 permit ::/0 ge 1 le
128
```

The following commands configure a distribution list that applies the filterOspfRoutesVe prefix list to routes pointing to virtual interface 10.

```
NetIron(config)# ipv6 router ospf
NetIron(config-ospf6-router)# distribute-list prefix-list filterOspfRoutes in ve
10
```

After this distribution list is configured, route 3015::/64, pointing to virtual interface 10, would be omitted from the OSPFv3 route table.

```
NetIron# show ipv6 ospf route

  Current Route count: 4
    Intra: 3 Inter: 0 External: 1 (Type1 0/Type2 1)
    Equal-cost multi-path: 0
    Destination                    Options   Area             Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 3001::/64                      --------- 0.0.0.1             0 0
    ::                             ve 10
*E2 3010::/64                      --------- 0.0.0.0            10 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 3020::/64                      --------- 0.0.0.0            10 0
    ::                             ve 11
*E2 6001:5000::/64                 --------- 0.0.0.0            10 0
    fe80::2e0:52ff:fe00:10         ve 10
```

Configuring an OSPFv3 distribution list using a route map as input

The following commands configure a route map that matches internal routes.

```
NetIron(config)# route-map allowInternalRoutes permit 10
NetIron(config-routemap allowInternalRoutes)# match route-type internal
```

Refer to 22, “Policy-Based Routing” for information on configuring route maps.

The following commands configure a distribution list that applies the allowInternalRoutes route map globally to OSPFv3 routes.

```
NetIron(config)# ipv6 router ospf
NetIron(config-ospf6-router)# distribute-list route-map allowinternalroutes in
```

Syntax: [no] distribute-list route-map <*name*> **in**

After this distribution list is configured, the internal routes would be included, and the external routes would be omitted from the OSPFv3 route table.

```
NetIron# show ipv6 ospf route

  Current Route count: 3
    Intra: 3 Inter: 0 External: 0 (Type1 0/Type2 0)
    Equal-cost multi-path: 0
    Destination                    Options   Area            Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 3001::/64                      --------- 0.0.0.1           0 0
    ::                             ve 10
*IA 3015::/64                      V6E---R-- 0.0.0.0          11 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 3020::/64                      --------- 0.0.0.0          10 0
    ::                             ve 11
```

Configuring default route origination

When the device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to automatically generate a default external route into an OSPFv3 routing domain. This feature is called “default route origination” or “default information origination.”

By default, the device does not advertise the default route into the OSPFv3 domain. If you want the device to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPF default route origination, the device advertises a type 5 default route that is flooded throughout the AS (except stub areas).

The device advertises the default route into OSPF even if OSPF route redistribution is not enabled, and even if the default route is learned through an IBGP neighbor.

NOTE
The device does not advertise the OSPF default route, regardless of other configuration parameters, unless you explicitly enable default route origination.

If default route origination is enabled and you disable it, the default route originated by the device is flushed. Default routes generated by other OSPF routers are not affected. If you re-enable the feature, the feature takes effect immediately and thus does not require you to reload the software.

For example, to create and advertise a default route with a metric of 2 and as a type 1 external route, enter the following command.

```
NetIron(config-ospf6-router)# default-information-originate always metric 2
metric-type type1
```

Syntax: [no] default-information-originate [always] [metric <*value*>**] [metric-type** <*type*>**]**

The **always** keyword originates a default route regardless of whether the device has learned a default route. This option is disabled by default.

The **metric** <*value*> parameter specifies a metric for the default route. If this option is not used, the value of the **default-metric** command is used for the route. For information about this command, refer to “Modifying default metric for routes redistributed into OSPF Version 3” on page 1817.

The **metric-type** <*type*> parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The <*type*> can be one of the following:

- 1 – Type 1 external route
- 2 – Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE
If you specify a metric and metric type, the values are used even if you do not use the always option.

To disable default route origination, enter the **no** form of the command.

Modifying Shortest Path First timers

The device uses the following timers when calculating the shortest path for OSPFv3 routes:

- **SPF delay** – When the device receives a topology change, the software waits before it starts a Shortest Path First (SPF) calculation. By default, the software waits 5 seconds. You can configure the SPF delay to a value from 0 – 65535 seconds. If you set the SPF delay to 0 seconds, the software immediately begins the SPF calculation after receiving a topology change.
- **SPF hold time** – The device waits a specific amount of time between consecutive SPF calculations. By default, it waits 10 seconds. You can configure the SPF hold time to a value from 0 – 65535 seconds. If you set the SPF hold time to 0 seconds, the software does not wait between consecutive SPF calculations.

You can set the SPF delay and hold time to lower values to cause the device to change to alternate paths more quickly if a route fails. Note that lower values for these parameters require more CPU processing time.

You can change one or both of the timers.

NOTE
If you want to change only one of the timers, for example, the SPF delay timer, you must specify the new value for this timer as well as the current value of the SPF hold timer, which you want to retain. The device does not accept only one timer value.

NOTE
If you configure SPF timers between 0-100, they will default to 0 and be displayed incorrectly in the running configuration.

To change the SPF delay to 10 seconds and the SPF hold to 20 seconds, enter the following command.

```
NetIron(config-ospf6-router)# timers spf 10 20
```

Syntax: timers spf <*delay*> <*hold-time*>

For the <*delay*> and <*hold-time*> parameters, specify a value from 0 – 65535 seconds.

To set the timers back to their default values, enter the **no** version of this command.

Modifying administrative distance

The device can learn about networks from various protocols, including BGP4+, IPv6 IS-IS, RIPng, and OSPFv3. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. By default, the administrative distance for OSPFv3 routes is 110.

The device selects one route over another based on the source of the route information. To do so, the device can use the administrative distances assigned to the sources. You can influence the device's decision by changing the default administrative distance for OSPFv3 routes.

Configuring administrative distance based on route type

You can configure a unique administrative distance for each type of OSPFv3 route. For example, you can use this feature to influence the device to prefer a static route over an OSPF inter-area route and to prefer OSPF intra-area routes to static routes.

The distance you specify influences the choice of routes when the device has multiple routes to the same network from different protocols. The device prefers the route with the lower administrative distance.

You can specify unique default administrative distances for the following OSPFv3 route types:

- Intra-area routes
- Inter-area routes
- External routes

The default for all of these OSPFv3 route types is 110.

NOTE
This feature does not influence the choice of routes within OSPFv3. For example, an OSPF intra-area route is always preferred over an OSPF inter-area route, even if the intra-area route's distance is greater than the inter-area route's distance.

For example, to change the default administrative distances for intra-area routes to 80, inter-area routes to 90, and external routes to 100, enter the following commands.

```
NetIron(config-ospf6-router)# distance intra-area 80
NetIron(config-ospf6-router)# distance inter-area 90
NetIron(config-ospf6-router)# distance external 100
```

Syntax: distance external | inter-area | intra-area *<distance>*

The external | inter-area | intra-area keywords specify the route type for which you are changing the default administrative distance.

The *<distance>* parameter specifies the new distance for the specified route type. You can specify a value from
1 – 255.

To reset the administrative distance of a route type to its system default, enter the no form of this command.

Configuring the OSPFv3 LSA pacing interval

The device paces OSPFv3 LSA refreshes by delaying the refreshes for a specified time interval instead of performing a refresh each time an individual LSA's refresh timer expires. The accumulated LSAs constitute a group, which the device refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the device refreshes an accumulated group of LSAs, is configurable to a range from 10 – 1800 seconds (30 minutes). The default is 240 seconds (four minutes). Thus, every four minutes, the device refreshes the group of accumulated LSAs and sends the group together in the same packets.

The pacing interval is inversely proportional to the number of LSAs the device is refreshing and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval enhances performance. If you have a very small database (40 – 100 LSAs), increasing the pacing interval to 10 – 20 minutes might enhance performance only slightly.

To change the OSPFv3 LSA pacing interval to two minutes (120 seconds), enter the following command.

```
NetIron(config)# ipv6 router ospf
NetIron(config-ospf6-router)# timers lsa-group-pacing 120
```

Syntax: [no] timers lsa-group-pacing <*seconds*>

The <*seconds*> parameter specifies the number of seconds and can be from 10 – 1800 (30 minutes). The default is 240 seconds (four minutes).

To restore the pacing interval to its default value, use the **no** form of the command.

Modifying exit overflow interval

If a database overflow condition occurs on the device, the device eliminates the condition by removing entries that originated on the device. The exit overflow interval allows you to set how often a device checks to see if the overflow condition has been eliminated. The default value is 0. If the configured value of the database overflow interval is 0, then the device never leaves the database overflow condition.

For example, to modify the exit overflow interval to 60 seconds, enter the following command.

```
NetIron(config-ospf6-router)# database-overflow-interval 60
```

Syntax: database-overflow-interval <*seconds*>

The <*seconds*> parameter can be a value from 0 – 86400 seconds (24 hours).

To reset the exit overflow interval to its system default, enter the **no** form of this command.

Modifying external link state database limit

By default, the link state database can hold a maximum of 2000 entries for external (type 5) LSAs. You can change the maximum number of entries from 500 – 8000. After changing this limit, make sure to save the running-config file and reload the software. The change does not take effect until you reload or reboot the software.

For example, to change the maximum number entries from the default of 2000 to 3000, enter the following command.

```
NetIron(config-ospf6-router)# external-lsdb-limit 3000
```

Syntax: **external-lsdb-limit** <*entries*>

The <*entries*> parameter can be a numerical value from 500 – 8000 seconds.

To reset the maximum number of entries to its system default, enter the **no** form of this command.

Modifying OSPFv3 interface defaults

OSPFv3 has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

You can modify the default values for the following OSPF interface parameters:

- **Cost:** Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps and 1000 Mbps (1 Gbps) links. The command syntax is **ipv6 ospf cost** <*number*>. The default cost is calculated by dividing 100 million by the bandwidth. For 10 Mbps links, the cost is 10. The cost for both 100 Mbps and 1000 Mbps links is 1, because the speed of 1000 Mbps was not in use at the time the OSPF cost formula was devised.
- **Dead-interval:** Indicates the number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The command syntax is **ipv6 ospf dead-interval** <*seconds*>. The value can be from 1 – 2147483647 seconds. The default is 40 seconds.
- **Hello-interval:** Represents the length of time between the transmission of hello packets. The command syntax is **ipv6 ospf hello-interval** <*seconds*>. The value can be from 1 – 65535 seconds. The default is 10 seconds.
- **Instance:** Indicates the number of OSPFv3 instances running on an interface. The command syntax is **ipv6 ospf instance** <*number*>. The value can be from 0 – 255. The default is 1.
- **MTU-ignore:** Allows you to disable a check that verifies the same MTU is used on an interface shared by neighbors. The command syntax is **ipv6 ospf mtu-ignore**. By default, the mismatch detection is enabled.
- **Network:** Allows you to configure the OSPF network type. The command syntax is **ipv6 ospf network** [**point-to-multipoint**]. The default setting of the parameter depends on the network type.
- **Passive:** When you configure an OSPF interface to be passive, that interface does not send or receive OSPF route updates. This option affects all IPv6 subnets configured on the interface. The command syntax is **ipv6 ospf passive**. By default, all OSPF interfaces are active and thus can send and receive OSPF route information. Since a passive interface does not send or receive route information, the interface is in effect a stub network.
- **Priority:** Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The command syntax is **ipv6 ospf priority** <*number*>. The value can be from 0 – 255. The default is 1. If you set the priority to 0, the router does not participate in DR and BDR election.
- **Retransmit-interval:** The time between retransmissions of LSAs to adjacent routers for an interface. The command syntax is **ipv6 ospf retransmit-interval** <*seconds*>. The value can be from 0 – 3600 seconds. The default is 5 seconds.

- **Transmit-delay:** The time it takes to transmit Link State Update packets on this interface. The command syntax is **ipv6 ospf transmit-delay** <*seconds*>. The range is 0 – 3600 seconds. The default is 1 second.

Disabling or reenabling event logging

OSPFv3 supports the generation of SNMP traps together with the logging of OSPFv3 events. The log-status change parameter controls the generation of all OSPFv3 logs and OSPFv3 SNMP traps. You can disable or re-enable the generation of SNMP traps and the logging of events related to OSPFv3, such as neighbor state changes and database overflow conditions. By default, the device logs these events and generates the SNMP traps.

To disable the logging of events and the generation of the SNMP traps, enter the following command.

```
NetIron(config-ospf6-router)# no log-status-change
```

Syntax: [no] log-status-change

To re-enable the logging of events, enter the following command.

```
NetIron(config-ospf6-router)# log-status-change
```

IPsec for OSPFv3

This section describes the current implementation of Internet Protocol Security (IPsec) for securing OSPFv3 traffic. For background information and configuration steps, refer to "Configuring IPsec for OSPFv3" on page 1827.

IPsec is available for OSPFv3 traffic only and only for packets that are "for-us." A for-us packet is addressed to one of the IPv6 addresses on the router or to an IPv6 multicast address. Packets that are just forwarded by the line card do not receive IPsec scrutiny.

In the current release, Dell supports the following components of IPsec for IPv6-addressed packets:

- Authentication through Encapsulating Security Payload (ESP) in transport mode
- HMAC-SHA1-96 as the authentication algorithm
- Manual configuration of keys
- Configurable rollover timer

IPsec can be enabled on the following logical entities:

- Interface
- Area
- Virtual link

With respect to traffic classes, this implementation of IPSec uses a single security association (SA) between the source and destination to support all traffic classes and so does not differentiate between the different classes of traffic that the DSCP bits define.

Instructions for configuring IPsec on these entities appear in "Configuring IPsec for OSPFv3" on page 1827.

IPsec on a virtual link is a global configuration. Interface and area IPsec configurations are more granular.

Among the entities that can have IPsec protection, the interfaces and areas can overlap. The interface IPsec configuration takes precedence over the area IPsec configuration when an area and an interface within that area use IPsec. Therefore, if you configure IPsec for an interface and an area configuration also exists that includes this interface, the interface's IPsec configuration is used by that interface. However, if you disable IPsec on an interface, IPsec is disabled on the interface even if the interface has its own, specific authentication. Refer to "Disabling IPsec on an interface" on page 1832.

For IPsec, the system generates two types of databases. The *security association database* (SAD) contains a security association for each interface or one global database for a virtual link. Even if IPsec is configured for an area, each interface that uses the area's IPsec still has its own security association in the SAD. Each SA in the SAD is a generated entry that is based on your specifications of an authentication protocol (ESP in the current release), destination address, and a security policy index (SPI). The SPI number is user-specified according to the network plan. Consideration for the SPI values to specify must apply to the whole network.

The system-generated security policy databases (SPDs) contain the security policies against which the system checks the for-us packets. For each for-us packet that has an ESP header, the applicable security policy in the security policy database (SPD) is checked to see if this packet complies with the policy. The IPsec task drops the non-compliant packets. Compliant packets continue on to the OSPFv3 task.

Configuring IPsec for OSPFv3

This section describes how to configure IPsec for an interface, area, and virtual link. It also describes how to change the key rollover timer if necessary and how to disable IPsec on a particular interface for special purposes.

By default, OSPFv3 IPSec authentication is disabled. The following IPsec parameters are configurable:

- ESP security protocol
- Authentication
- HMAC-SHA1-96 authentication algorithm
- Security parameter index (SPI)
- A 40-character key using hexadecimal characters
- An option for not encrypting the keyword when it appears in **show** command output
- Key rollover timer

NOTE
In the current release, certain keyword parameters must be entered even though only one keyword choice is possible for that parameter. For example, the only authentication algorithm in the current release is HMAC-SHA1-96, but you must nevertheless enter the keyword for this algorithm. Also, ESP currently is the only authentication protocol, but you must still enter the **esp** keyword. This section describes all keywords.

General considerations

The IPsec component generates security associations and security policies based on certain user-specified parameters. The parameters are described with the syntax of each command in this section and also pointed out in the section with the **show** command examples, "IPsec examples" on page 1855. User-specified parameters and their relation to system-generated values are as follows:

- **Security association:** based on your entries for *security policy index* (SPI), *destination address*, and *security protocol* (currently ESP), the system creates a security association for each interface or virtual link.
- **Security policy database:** based on your entries for SPI, *source address*, *destination addresses*, and *security protocol*, the system creates a security policy database for each interface or virtual link.
- You can configure the same SPI and key on multiple interfaces and areas, but they still have unique IPsec configurations because the SA and policies are added to each separate security policy database (SPD) that is associated with a particular interface. If you configure an SA with the same SPI in multiple places, the rest of the parameters associated with the SA—such as key, crypto algorithm, and security protocol, and so on—must match. If the system detects a mismatch, it displays an error message.
- IPSec authentication for OSPFv3 requires the use of multiple SPDs, one for each interface. A virtual link has a separate, global SPD. The authentication configuration on a virtual link must be different from the authentication configuration for an area or interface, as required by RFC4552. The interface number is used to generate a non-zero security policy database identifier (SPDID), but for the global SPD for a virtual link, the system-generated SPDID is always zero. As a hypothetical example, the SPD for interface eth 1/1 might have the system-generated SPDID of 1, and so on.
- If you change an existing key, you must also specify a different SPI value. For example, in an interface context where you intend to change a key, you must type a different SPI value—which occurs before the key parameter on the command line—before you type the new key. The example in "Configuring IPsec for OSPFv3"illustrates this requirement.
- The old key is active for twice the current configured key-rollover-interval for the inbound direction. In the outbound direction, the old key remains active for a duration equal to the key-rollover-interval. If the key-rollover-interval is set to 0, the new key immediately takes effect for both directions. For a description of the key-rollover-interval, refer to the "Changing the key rollover timer" on page 1832section.

Interface and area IPsec considerations

This section describes the precedence of interface and area IPsec configurations.

If you configure an interface IPsec by using the **ipv6 ospf authentication** command in the context of a specific interface, that interface's IPsec configuration overrides the area configuration of IPsec.

If you configure IPsec for an area, all interfaces that utilize the area-wide IPsec (where interface-specific IPsec is not configured) nevertheles receive an SPD entry (and SPDID number) that is unique for the interface.

The area-wide SPI that you specify is a constant for all interfaces in the area that use the area IPsec, but the use of different interfaces results in an SPDID and an SA that are unique to each interface. (Recall from "IPsec for OSPFv3" on page 1826 that the security policy database depends partly on the source IP address, so a unique SPD for each interface results.)

Considerations for IPsec on virtual links

The IPsec configuration for a virtual link is global, so only one security association database and one security policy database exist for virtual links if you choose to configure IPsec for virtual links.

The virtual link IPsec SAs and policies are added to all interfaces of the transit area for the outbound direction. For the inbound direction, IPsec SAs and policies for virtual links are added to the global database.

NOTE
The security association (SA), security protocol index (SPI), security protocol database (SPD), and key have mutual dependencies, as the subsections that follow describe.

Specifying the key rollover timer

Configuration changes for authentication takes effect in a controlled manner through the key rollover procedure as specified in RFC 4552, Section 10.1. The key rollover timer controls the timing of the configuration changeover. The key rollover timer can be configured in the IPv6 router OSPF context, as the following example illustrates.

```
NetIron(config-ospf6-router)#key-rollover-interval 200
```

Syntax: **key-rollover-interval** <*time*>

The range for the key-rollover-interval is 0 – 14400 seconds. The default is 300 seconds.

Configuring IPsec on a interface

For IPsec to work, the IPsec configuration must be the same on all the routers to which an interface connects.

For multicast, IPsec does not need or use a specific destination address—the destination address is “do not care,” and this status is reflected by the lone pair of colons (::) for destination address in the **show** command output.

To configure IPsec on an interface, proceed as in the following example.

NOTE
The IPsec configuration for an interface applies to the inbound and outbound directions. Also, the same authentication parameters must be used by all routers on the network to which the interface is connected, as described in section 7 of RFC 4552.

```
NetIron(config-if-e10000-1/2)#ipv6 ospf auth ipsec spi 429496795 esp sha1
abcdef12345678900987654321fedcba12345678
```

Syntax: **[no] ipv6 ospf authentication ipsec spi** <*spinum*> **esp sha1 [no-encrypt]** <*key*>

The **no** form of this command deletes IPsec from the interface.

The **ipv6** command is available in the configuration interface context for a specific interface.

The **ospf** keyword identifies OSPFv3 as the protocol to receive IPsec security.

The **authentication** keyword enables authentication.

The **ipsec** keyword specifies IPsec as the authentication protocol.

The **spi** keyword and the <*spinum*> variable specify the security parameter that points to the security association. The near-end and far-end values for spinum must be the same. The range for <*spinum*> is decimal 256 – 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that when you display the IPsec configuration, the key is displayed in its unencrypted form and also saved as unencrypted.

The <*key*> variable must be 40 hexadecimal characters. To change an existing key, you must also specify a different SPI value. You cannot just change the key without also specifying a different SPI, too. For example, in an interface context where you intend to change a key, you must type a different SPI value—which occurs before the key parameter on the command line—before you type the new key. The example in "Configuring IPsec for OSPFv3"illustrates this requirement.

If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- encryptb64 = the key string uses proprietary base64 crytographic 2-way algorithm

This example results in the configuration shown in the screen output that follows. Note that because the optional **no-encrypt** keyword was omitted, the display of the key has the encrypted form by default.

```
interface ethernet 1/2
 enable
 ip address 40.3.3.1/8
 ipv6 address 40:3:3::1/64
 ipv6 ospf area 1
 ipv6 ospf authentication ipsec spi 429496795 esp sha1 encryptb64
$ITJkQG5HWnw4M09tWVd
```

Configuring IPsec for an area

This application of the **area** command (for IPsec) applies to all of the interfaces that belong to an area unless an interface has its own IPsec configuration. (As described in "Disabling IPsec on an interface" on page 1832, the interface IPsec can be operationally disabled if necessary.) To configure IPsec for an area in the IPv6 router OSPF context, proceed as in the following example.

```
NetIron(config-ospf6-router)#area 2 auth ipsec spi 400 esp sha1
abcef12345678901234fedcba098765432109876
```

Syntax: **area** <*area-id*> **authentication ipsec spi** <*spinum*> **esp sha1 [no-encrypt]** <*key*>

The **no** form of this command deletes IPsec from the area.

The **area** command and the <*area-id*> variable specify the area for this IPsec configuration. The <*area-id*> can be an integer in the range 0 – 2,147,483,647 or have the format of an IP address.

The **authentication** keyword specifies that the function to specify for the area is packet authentication.

The **ipsec** keyword specifies that IPsec is the protocol that authenticates the packets.

The **spi** keyword and the <*spinum*> variable specify the index that points to the security association. The near-end and far-end values for spinum must be the same. The range for <*spinum*> is decimal 256 – 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that the 40-character key is not encrypted upon either its entry or its display. The key must be 40 hexadecimal characters.

If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- encryptb64 = the key string uses proprietary base64 crytographic 2-way algorithm

The configuration in the preceding example results in the configuration for area 2 that is illustrated in the following.

```
ipv6 router ospf
 area 0
 area 1
 area 2
 area 2 auth ipsec spi 400 esp sha1 abcef12345678901234fedcba098765432109876
```

Configuring IPsec for a virtual link

IPsec on a virtual link has a global configuration.

To configure IPsec on a virtual link, enter the IPv6 router OSPF context of the CLI and proceed as the following example illustrates. (Note the **no-encrypt** option in this example.)

```
NetIron(config-ospf6-router)#area 1 vir 2.2.2.2 auth ipsec spi 360 esp sha1
no-encrypt 1234567890098765432112345678990987654321
```

Syntax: [no] **area** <*area-id*> **virtual** <*nbrid*> **authentication ipsec spi** <*spinum*> **esp sha1** [**no-encrypt**] <*key*>

The **no** form of this command deletes IPsec from the virtual link.

The **area** command and the <*area-id*> variable specify the area is to be configured. The <*area-id*> can be an integer in the range 0 – 2,147,483,647 or have the format of an IP address.

The **virtual** keyword indicates that this configuration applies to the virtual link identified by the subsequent variable <*nbrid*>. The variable <*nbrid*> is in dotted decimal notation of an IP address.

The **authentication** keyword specifies that the function to specify for the area is packet authentication.

The **ipsec** keyword specifies that IPsec is the protocol that authenticates the packets.

The **spi** keyword and the <*spinum*> variable specify the index that points to the security association. The near-end and far-end values for spinum must be the same. The range for <*spinum*> is decimal 256 – 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that the 40-character key is not encrypted in **show** command displays. If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- encryptb64 = the key string uses proprietary base64 crytographic 2-way algorithm

This example results in the following configuration.

```
area 1 virtual-link 2.2.2.2
area 1 virtual-link 2.2.2.2 authentication ipsec spi 360 esp sha1 no-encrypt 12
3456789009876543211234567890987654321
```

Disabling IPsec on an interface

For the purpose of troubleshooting, you can operationally disable IPsec on an interface by using the **ipv6 ospf authentication ipsec disable** command in the CLI context of a specific interface. This command disables IPsec on the interface whether its IPsec configuration is the area's IPsec configuration or is specific to that interface. The output of the **show ipv6 ospf interface command** shows the current setting for the disable command.

To disable IPsec on an interface, go to the CLI context of the interface and proceed as in the following example.

```
NetIron(config-if-e10000-1/2)#ipv6 ospf auth ipsec disable
```

Syntax: [no] ipv6 ospf authentication ipsec disable

The **no** form of this command restores the area and interface-specific IPsec operation.

Changing the key rollover timer

Configuration changes for authentication takes effect in a controlled manner through the key rollover procedure as specified in RFC 4552, Section 10.1. The key rollover timer controls the timing of the configuration changeover. The key rollover timer can be configured in the IPv6 router OSPF context, as the following example illustrates.

```
NetIron(config-ospf6-router)#key-rollover-interval 200
```

Syntax: key-rollover-interval <*time*>

The range for the key-rollover-interval is 0 – 14400 seconds. The default is 300 seconds.

Clearing IPsec statistics

This section describes the **clear ipsec statistics** command for clearing statistics related to IPsec. The command resets to 0 the counters (which you can view as a part of IPSecurity Packet Statistics). The counters hold IPsec packet statistics and IPsec error statistics. The following example illustrates the **show ipsec statistics** output.

```
NetIron#show ipsec statistics
                    IPSecurity Statistics
secEspCurrentInboundSAs 1          ipsecEspTotalInboundSAs:  2
secEspCurrentOutboundSA 1          ipsecEspTotalOutboundSAs: 2
                  IPSecurity Packet Statistics
secEspTotalInPkts:      20         ipsecEspTotalInPktsDrop:  0
secEspTotalOutPkts:     84
                  IPSecurity Error Statistics
secAuthenticationErrors 0
secReplayErrors:        0          ipsecPolicyErrors:        13
secOtherReceiveErrors:  0          ipsecSendErrors:          0
secUnknownSpiErrors:    0
```

To clear the statistics, enter the **clear ipsec statistics** command as in the following example.

```
NetIron#clear ipsec statistics
```

Syntax: clear ipsec statistics

This command takes no parameters.

Displaying OSPFv3 information

You can display the information for the following OSPFv3 parameters:

- Areas
- Link state databases
- Interfaces
- Memory usage
- Neighbors
- Redistributed routes
- Routes
- SPF
- Virtual links
- Virtual neighbors
- IPsec

General OSPF configuration information

To indicate whether the router is operating as ASBR or not, enter the following command at any CLI level.

```
NetIron#show ipv6 ospf
OSPFv3 Process number 0 with Router ID 0x01010101(1.1.1.1)
Running 0 days 0 hours 1 minutes 53 seconds
Number of AS scoped LSAs is 3
Sum of AS scoped LSAs Checksum is fabdd4de
External LSA Limit is 250000
Database Overflow Interval is 10
Database Overflow State is NOT OVERFLOWED
Route calculation executed 0 times
Pending outgoing LSA count 0
Authentication key rollover interval 30 seconds
```

```
Number of areas in this router is 4
Router is operating as ABR
Router is operating as ASBR, Redistribute: CONNECTED
High Priority Message Queue Full count: 0
BFD is disabled
```

The output of the **show ipv6 ospf** command will indicate if the router is operating as ASBR. If the router is not operating as ASBR, then there will be no information about redistribution in the output.

Displaying OSPFv3 area information

To display global OSPFv3 area information for the device, enter the following command at any CLI level.

```
NetIron# show ipv6 ospf area
Area 0:
  Interface attached to this area: loopback 2 ethe 3/2 tunnel 2
  Number of Area scoped LSAs is 6
  Statistics of Area 0:
    SPF algorithm executed 16 times
    SPF last updated: 335256 sec ago
    Current SPF node count: 3
      Router: 2 Network: 1
      Maximum of Hop count to nodes: 2
...
```

Syntax: show ipv6 ospf area [<*area-id*>]

You can specify the <*area-id*> parameter in the following formats:

- As an IPv4 address, for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

The <*area-id*> parameter restricts the display to the specified OSPF area.

This display shows the following information.

TABLE 317 OSPFv3 area information fields

This field...	Displays...
Area	The area number.
Interface attached to this area	The router interfaces attached to the area.
Number of Area scoped LSAs is *N*	Number of LSAs (*N*) with a scope of the specified area.
SPF algorithm executed is *N*	The number of times (*N*) the OSPF Shortest Path First (SPF) algorithm is executed within the area.
SPF last updated	The interval in seconds that the SPF algorithm was last executed within the area.
Current SPF node count	The current number of SPF nodes in the area.
Router	Number of router LSAs in the area.
Network	Number of network LSAs in the area.
Indx	The row number of the entry in the router's OSPF area table.
Statistics of Area	The number of the area whose statistics are displayed.
Maximum hop count to nodes.	The maximum number of hop counts to an SPF node within the area.

Displaying OSPFv3 database information

You can display a summary of the device's link state database or detailed information about a specified LSA type.

To display a summary of a device's link state database, enter the following command at any CLI level.

```
NetIron# show ipv6 ospf database
LSA Key - Rtr:Router Net:Network Inap:InterPrefix Inar:InterRouter
          Extn:ASExternal Grp:GroupMembership Typ7:Type7 Link:Link
          Iap:IntraPrefix
Area ID         Type LSID       Adv Rtr         Seq(Hex) Age  Cksum Len
1               Link 1          2.2.2.2         80000002 998  d1b7  56
1               Link 1          1.1.1.1         80000002 996  dbb2  56
2               Link 21         3.3.3.3         80000002 998  76b1  56
2               Link 21         1.1.1.1         80000002 996  8aa7  56
1               Rtr  0          2.2.2.2         80000006 1392 e423  40
1               Rtr  0          1.1.1.1         80000008 850  0521  24
1               Net  1          2.2.2.2         80000002 1392 5eb4  32
1               Inap 1          1.1.1.1         80000002 997  2eb0  36
1               Inap 4          1.1.1.1         80000002 996  9723  36
```

Syntax: **show ipv6 ospf database [advrtr** <*ipv4-address*> **| as-external[advrtr** <*ipv4-address*> **| link-id** <*number*> **] | extensive | inter-prefix [advrtr** <*ipv4-address*> **| link-id** <*number*> **] | inter-router[advrtr** <*ipv4-address*> **| link-id** <*number*> **] | intra-prefix [advrtr** <*ipv4-address*> **| link-id** <*number*> **] | link [advrtr** <*ipv4-address*> **| link-id** <*number*> **] | link-id** <*number*> **| network [advrtr** <*ipv4-address*> **| link-id** <*number*> **] | router [advrtr** <*ipv4-address*> **| link-id** <*number*> **]]**

The **advrtr** <*ipv4-address*> parameter displays detailed information about the LSAs for a specified advertising router only.

The **as-external** keyword displays detailed information about the AS externals LSAs only.

The **extensive** keyword displays detailed information about all LSAs in the database.

The **inter-prefix** keyword displays detailed information about the inter-area prefix LSAs only.

The **inter-router** keyword displays detailed information about the inter-area router LSAs only.

The **intra-prefix** keyword displays detailed information about the intra-area prefix LSAs only.

The **link** keyword displays detailed information about the link LSAs only.

The **link-id** <*number*> parameter displays detailed information about the specified link LSAs only.

The **network** <*number*> displays detailed information about the network LSAs only.

The **router** <*number*> displays detailed information about the router LSAs only.

The **scope** <*area-id*> parameter displays detailed information about the LSAs for a specified area, AS, or link.

This display shows the following information.

TABLE 318 OSPFv3 database summary fields

This field...	Displays...
Area ID	The OSPF area in which the device resides.
Type	Type of LSA. LSA types can be the following: • Rtr – Router LSAs (Type 1). • Net – Network LSAs (Type 2). • Inap – Inter-area prefix LSAs for ABRs (Type 3). • Inar – Inter-area router LSAs for ASBRs (Type 4). • Extn – AS external LSAs (Type 5). • Link – Link LSAs (Type 8). • Iap – Intra-area prefix LSAs (Type 9).
LS ID	The ID of LSA in Decimal.
Adv Rtr	The device that advertised the route.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps it with a sequence number to enable the device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA, in seconds.
Chksum	A checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.
Len	The length, in bytes, of the LSA.

For example, to display detailed information about all LSAs in the database, enter the following command at any CLI level.

```
NetIron# show ipv6 ospf database extensive
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Link 00000031 1.1.1.1         80000001 35   6db9   56
     Router Priority: 1
     Options: V6E---R--
     LinkLocal Address: fe80::1
     Number of Prefix: 1
     Prefix Options:
     Prefix: 3002::/64
    ...
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Iap  00000159 223.223.223.223 800000ab 357  946b   56
     Number of Prefix: 2
     Referenced LS Type: Network
     Referenced LS ID: 00000159
     Referenced Advertising Router: 223.223.223.223
     Prefix Options:  Metric: 0
     Prefix: 2000:4::/64
     Prefix Options:  Metric: 0
     Prefix: 2002:c0a8:46a::/64
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Rtr  00000039 223.223.223.223 800000b1 355  8f2d   40
  Capability Bits: --E-
  Options: V6E---R--
  Type: Transit Metric: 1
  Interface ID: 00000058  Neighbor Interface ID: 00000058
  Neighbor Router ID: 223.223.223.223
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Net  000001f4 223.223.223.223 800000ab 346  190a   32
     Options: V6E---R--
     Attached Router: 223.223.223.223
     Attached Router: 1.1.1.1
...
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
N/A             Extn 000001df 223.223.223.223 800000af 368  0aa8   32
     Bits: E
     Metric: 00000001
     Prefix Options:
     Referenced LSType: 0
     Prefix: 2002::/16
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
1               Inap 0000011d 10.1.1.188      80000001 124  25de   36
     Metric: 2
     Prefix Options:
     Prefix: 2000:2:2::/64
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Inar 0000005b 10.1.1.198      80000001 990  dbad   32
     Options: V6E---R--
     Metric: 1
     Destination Router ID:10.1.1.188
```

NOTE

Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

The fields that display depend upon the LSA type as shown in the following.

TABLE 319 OSPFv3 detailed database information fields

This field...	Displays...
Router LSA (Type 1) (Rtr) Fields	
Capability Bits	A bit that indicates the capability of the device. The bit can be set to one of the following: B – The device is an area border router. E – The device is an AS boundary router. V – The device is a virtual link endpoint. W – The device is a wildcard multicast receiver.
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The device handles demand circuits.
Type	The type of interface. Possible types can be the following: Point-to-point – A point-to-point connection to another router. Transit – A connection to a transit network. Virtual link – A connection to a virtual link.
Metric	The cost of using this router interface for outbound traffic.
Interface ID	The ID assigned to the router interface.
Neighbor Interface ID	The interface ID that the neighboring router has been advertising in hello packets sent on the attached link.
Neighbor Router ID	The router ID (IPv4 address) of the neighboring router that advertised the route. (By default, the router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.)

TABLE 319 OSPFv3 detailed database information fields (Continued)

This field...	Displays...
Network LSA (Type 2) (Net) Fields	
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The device handles demand circuits.
Attached Router	The address of the neighboring router that advertised the route.
Inter-Area Prefix LSA (Type 3) (Inap) Fields	
Metric	The cost of the route.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Prefix	The IPv6 prefix included in the LSA.
Inter-Area Router LSA (Type 4) (Inar) Fields	
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The device handles demand circuits.
Metric	The cost of the route.
Destination Router ID	The ID of the router described in the LSA.
AS External LSA (Type 5) (Extn) Fields	
Bits	The bit can be set to one of the following: • E – If bit E is set, a Type 2 external metric. If bit E is zero, a Type 1 external metric. • F – A forwarding address is included in the LSA. • T – An external route tag is included in the LSA.
Metric	The cost of this route, which depends on bit E.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Referenced LS Type	If non-zero, an LSA with this LS type is associated with the LSA.
Prefix	The IPv6 prefix included in the LSA.
Link LSA (Type 8) (Link) Fields	
Router Priority	The router priority of the interface attaching the originating router to the link.
Options	The set of options bits that the router would like set in the network LSA that will be originated for the link.
Link Local Address	The originating router's link-local interface address on the link.
Number of Prefix	The number of IPv6 address prefixes contained in the LSA.

TABLE 319 OSPFv3 detailed database information fields (Continued)

This field...	Displays...
Prefix Options	An 8-bit field of capabilities that serve as input to various routing calculations: • NU – The prefix is excluded from IPv6 unicast calculations. • LA – The prefix is an IPv6 interface address of the advertising router. • MC – The prefix is included in IPv6 multicast routing calculations. • P – NSSA area prefixes are readvertised at the NSSA area border.
Prefix	The IPv6 prefix included in the LSA.
Intra-Area Prefix LSAs (Type 9) (Iap) Fields	
Number of Prefix	The number of prefixes included in the LSA.
Referenced LS Type, Referenced LS ID	Identifies the router-LSA or network-LSA with which the IPv6 address prefixes are associated.
Referenced Advertising Router	The address of the neighboring router that advertised the route.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Metric	The cost of using the advertised prefix.
Prefix	The IPv6 prefix included in the LSA.
Number of Prefix	The number of prefixes included in the LSA.

Displaying IPv6 interface information

You can use the following command to display a summary of IPv6 Interface information.

```
NetIron#show ipv6 interface
    Routing Protocols : R - RIP  O - OSPF I - ISIS
    Interface  Status/Protocol  Routing  IPv6 Address
    eth 1/7      up/up          I         fe80::20c:dbff:fef4:7106
                                          bb::21/64
                                          2001::2/64
    pos 4/2    down/down        I         4001::15/64
    loopback 1 up/up                      fe80::20c:dbff:fef4:7100
                                          4004::1/128
```

Syntax: show ipv6 interface [ethernet *<port>* **| pos** *<port>* **| loopback** *<number>* **| tunnel** *<number>* **| ve** *<number>* **]**

The **ethernet | pos | loopback | tunnel | ve** parameter specifies the interface for which to display information. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the interface.

This display shows the following information.

TABLE 320 Summary of IPv6 interface information

This field...	Displays...
Interface	The interface type, and the port number or number of the interface.
Status or Protocol	The status of the link and the protocol. Possible status include the following: • Up. • Down.

TABLE 320 Summary of IPv6 interface information (Continued)

This field...	Displays...
Routing	The routing protocol enabled on the interface. Possible values include: • R – RIP • O – OSPF • I – ISIS
IPv6 Address	The link local IPv6 address and any other IPv6 addresses configured for the interface.

Displaying IPv6 OSPFv3 interface information

IPv6 Interface information can be displayed in either a brief or full mode. The following sections describe the command to display these modes and the resulting output:

- Displaying IPv6 OSPFv3 Interface Information in Brief Mode
- Displaying IPv6 OSPFv3 Interface Information in Full Mode

Displaying IPv6 OSPFv3 interface information in brief mode

You can use the following command to display a summary of IPv6 Interface information.

```
NetIron# show ipv6 ospf interface brief
Interface  Area           Status  Type Cost  State     Nbrs (F/C)
eth 1/1    0              up      BCST 1     DROther   1/1
loopback 1 0              up      BCST 1     Loopback  0/0
```

Syntax: show ipv6 ospf interface brief

This display shows the following information.

TABLE 321 Summary of OSPFv3 interface brief information

This field...	Displays...
Interface	The interface type, and the port number or number of the interface.
Area	The OSPF area configured on the interface.
Status	The status of the link and the protocol. Possible status include the following: • Up. • Down.
Type	The type of OSPFv3 circuit running on the interface. Possible types include the following: • BCST- Broadcast interface type • P2P- Point-to-point interface type • UNK- The interface type is not known at this time
Cost	The overhead required to send a packet across an interface.

TABLE 321 Summary of OSPFv3 interface brief information (Continued)

This field...	Displays...
State	The state of the interface. Possible states include the following: • DR – The interface is functioning as the Designated Router for OSPFv3. • BDR – The interface is functioning as the Backup Designated Router for OSPFv3. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.
Nbrs (F/C)	The number of adjacent neighbor routers. The number to the left of the "/" are the neighbor routers that are fully adjacent and the number to the right represents all adjacent neighbor routers.

Displaying IPv6 OSPFv3 interface information in full mode

You can display detailed information about all OSPFv3 interfaces by using the **show ipv6 ospf interface** command, as the following truncated example illustrates.

```
NetIron#show ipv6 ospf interface
eth 1/3 is down, type BROADCAST
   Interface is disabled
eth 1/8 is up, type BROADCAST
  IPv6 Address:
      2100:18:18:18:18::1/64
      2100:18:18:18::/64
  Instance ID 255, Router ID 1.1.1.1
  Area ID 1, Cost 1
  State BDR, Transmit Delay 1 sec, Priority 1
  Timer intervals :
    Hello 10, Hello Jitter 10  Dead 40, Retransmit 5
  Authentication: Enabled
   KeyRolloverTime(sec): Configured: 30 Current: 0
   KeyRolloverState: NotActive
   Outbound: SPI:121212, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
   Inbound: SPI:121212, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
  DR:2.2.2.2 BDR:1.1.1.1  Number of I/F scoped LSAs is 2
  DRElection:     1 times, DelayedLSAck:    83 times
  Neighbor Count = 1,   Adjacent Neighbor Count= 1
     Neighbor:
      2.2.2.2 (DR)
    Statistics of interface eth 1/8:
      Type     tx         rx         tx-byte    rx-byte
      Unknown  0          0          0          0
      Hello    1415       1408       56592      56320
      DbDesc   3          3          804        804
      LSReq    1          1          28         28
      LSUpdate 193        121        15616      9720
      LSAck    85         109        4840       4924
      OSPF messages dropped,no authentication: 0
eth 2/2 is up, type POINT-TO-POINT
  IPv6 Address:
      2222:22:22:22::1/64
      2222:22:22:22::/64
      2222:202:202:202::1/64
      2222:202:202:202::/64
  Instance ID 0, Router ID 1.1.1.1
  Area ID 100, Cost 1
  State P2P, Transmit Delay 1 sec, Priority 1
  Timer intervals:
    Hello 10, Hello Jitter 10  Dead 40, Retransmit 5
  Authentication: Enabled
   KeyRolloverTime(sec): Configured: 30 Current: 0
   KeyRolloverState: NotActive
   Outbound: SPI:11022, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
   Inbound: SPI:11022, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
  DR:0.0.0.0 BDR:0.0.0.0  Number of I/F scoped LSAs is 2
  ....................................
```

You can display detailed OSPFv3 information about a specific interface using the following command at any level of the CLI.

Syntax: show ipv6 ospf interface [ethernet <*slot/port*> **| pos** <*slot/port*> **| loopback** <*number*> **| tunnel** <*number*> **| ve** <*number*> **]**

This display shows the following information.

TABLE 322 Detailed OSPFv3 interface information

This field...	Displays...
Interface status	The status of the interface (POS or Ethernet). Possible status includes the following: • Up. • Down.
Type	The type of OSPFv3 circuit running on the interface. Possible types include the following: • BROADCAST • POINT TO POINT UNKNOWN • POINT TO POINT
IPv6 Address	The IPv6 address(es) assigned to the interface.
Instance ID	An identifier for an instance of OSPFv3.
Router ID	The IPv4 address of the device. By default, the router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Area ID	The IPv4 address or numerical value of the area in which the interface belongs.
Cost	The overhead required to send a packet through the interface.
State	The state of the interface. Possible states include the following: • DR – The interface is functioning as the Designated Router for OSPFv3. • BDR – The interface is functioning as the Backup Designated Router for OSPFv3. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.
Transmit delay	The amount of time, in seconds, it takes to transmit Link State Updates packets on the interface.
Priority	The priority used when selecting the DR and the BDR. If the priority is 0, the interface does not participate in the DR and BDR election.
Timer intervals	The interval, in seconds, of the hello-interval, dead-interval, and retransmit-interval timers.
DR	The router ID (IPv4 address) of the DR.

TABLE 322 Detailed OSPFv3 interface information (Continued)

This field...	Displays...
BDR	The router ID (IPv4 address) of the BDR.
Number of I/F scoped LSAs	The number of interface LSAs scoped for a specified area, AS, or link.
DR Election	The number of times the DR election occurred.
Delayed LSA Ack	The number of the times the interface sent a delayed LSA acknowledgement.
Neighbor Count	The number of neighbors to which the interface is connected.
Adjacent Neighbor Count	The number of neighbors with which the interface has formed an active adjacency.
Neighbor	The router ID (IPv4 address) of the neighbor. This field also identifies the neighbor as a DR or BDR, if appropriate.
Interface statistics	The following statistics are provided for the interface: • Unknown – The number of Unknown packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Unknown packets. • Hello – The number of Hello packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Hello packets. • DbDesc – The number of Database Description packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Database Description packets. • LSReq – The number of link-state requests transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state requests. • LSUpdate – The number of link-state updates transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state requests. • LSAck – The number of link-state acknowledgements transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state acknowledgements.

Displaying OSPFv3 memory usage

To display information about OSPFv3 memory usage, enter the following command at any level of the CLI.

```
NetIron# show ipv6 ospf memory
Total Static Memory Allocated : 5829 bytes
Total Dynamic Memory Allocated : 0 bytes
Memory Type               Size       Allocated  Max-alloc  Alloc-Fails
MTYPE_OSPF6_TOP           0          0          0          0
MTYPE_OSPF6_LSA_HDR       0          0          0          0
MTYPE_OSPF6_RMAP_COMPILED 0          0          0          0
MTYPE_OSPF6_OTHER         0          0          0          0
MTYPE_THREAD_MASTER       0          0          0          0
MTYPE_OSPF6_AREA          0          0          0          0
MTYPE_OSPF6_AREA_RANGE    0          0          0          0
MTYPE_OSPF6_SUMMARY_ADDRE 0          0          0          0
MTYPE_OSPF6_IF            0          0          0          0
MTYPE_OSPF6_NEIGHBOR      0          0          0          0
MTYPE_OSPF6_ROUTE_NODE    0          0          0          0
MTYPE_OSPF6_ROUTE_INFO    0          0          0          0
MTYPE_OSPF6_PREFIX        0          0          0          0
MTYPE_OSPF6_LSA           0          0          0          0
MTYPE_OSPF6_VERTEX        0          0          0          0
MTYPE_OSPF6_SPFTREE       0          0          0          0
MTYPE_OSPF6_NEXTHOP       0          0          0          0
MTYPE_OSPF6_EXTERNAL_INFO 0          0          0          0
MTYPE_THREAD              0          0          0          0
```

Syntax: show ipv6 ospf memory

This display shows the following information.

TABLE 323 OSPFv3 memory usage information

This field...	Displays...
Total Static Memory Allocated	A summary of the amount of static memory allocated, in bytes, to OSPFv3.
Total Dynamic Memory Allocated	A summary of the amount of dynamic memory allocated, in bytes, to OSPFv3.
Memory Type	The type of memory used by OSPFv3. (This information is for use by Dell technical support in case of a problem.)
Size	The size of a memory type.
Allocated	The amount of memory currently allocated to a memory type.
Max-alloc	The maximum amount of memory that was allocated to a memory type.
Alloc-Fails	The number of times an attempt to allocate memory to a memory type failed.

Displaying OSPFv3 neighbor information

You can display a summary of OSPFv3 neighbor information for the device or detailed information about a specified neighbor.

To display a summary of OSPFv3 neighbor information for the device, enter the following command at any CLI level.

```
NetIron# show ipv6 ospf neighbor
RouterID        Pri State    DR              BDR             Interface[State]
1.1.1.1           1 Full     223.223.223.223 1.1.1.1         ethe 3/2   [DR]
```

Syntax: show ipv6 ospf neighbor [router-id <*ipv4-address*>**]**

The **router-id** <*ipv4-address*> parameter displays only the neighbor entries for the specified router.

This display shows the following information.

TABLE 324 Summary of OSPFv3 neighbor information

Field	Description
Router ID	The IPv4 address of the neighbor. By default, the router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Pri	The OSPFv3 priority of the neighbor. The priority is used during election of the DR and BDR.
State	The state between the device and the neighbor. The state can be one of the following: • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.
Interface [State]	The interface through which the router is connected to the neighbor. The state of the interface can be one of the following: • DR – The interface is functioning as the Designated Router for OSPFv3. • BDR – The interface is functioning as the Backup Designated Router for OSPFv3. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.

For example, to display detailed information about a neighbor with the router ID of 1.1.1.1, enter the following command at any CLI level.

```
NetIron# show ipv6 ospf neighbor router-id 3.3.3.3
RouterID        Pri State    DR              BDR             Interface[State]
3.3.3.3           1 Full     3.3.3.3         1.1.1.1         ve 10    [BDR]
DbDesc bit for this neighbor: --s
Nbr Ifindex of this router: 1
Nbr DRDecision: DR 3.3.3.3, BDR 1.1.1.1
Last received DbDesc: opt:xxx ifmtu:0 bit:--s seqnum:0
Number of LSAs in DbDesc retransmitting: 0
Number of LSAs in SummaryList: 0
Number of LSAs in RequestList: 0
Number of LSAs in RetransList: 0
SeqnumMismatch 0 times, BadLSReq 0 times
OnewayReceived 0 times, InactivityTimer 0 times
DbDescRetrans 0 times, LSReqRetrans 0 times
LSUpdateRetrans 1 times
LSAReceived 12 times, LSUpdateReceived 6 times
```

This display shows the following information.

TABLE 325 Detailed OSPFv3 neighbor information

Field	Description
Router ID	For information about this field, refer to Table 324 on page 1847.
Pri	For information about this field, refer to Table 324 on page 1847.
State	For information about this field, refer to Table 324 on page 1847.
DR	For information about this field, refer to Table 324 on page 1847.
BDR	For information about this field, refer to Table 324 on page 1847.
Interface [State]	For information about this field, refer to Table 324 on page 1847.
DbDesc bit...	The Database Description packet, which includes 3 bits of information: • The first bit can be “i” or “-”. “i” indicates the inet bit is set. “-” indicates the inet bit is not set. • The second bit can be “m” or “-”. “m” indicates the more bit is set. “-” indicates the more bit is not set. • The third bit can be “m” or “s”. An “m” indicates the master. An “s” indicates standby.
Index	The ID of the LSA from which the neighbor learned of the router.
DR Decision	The router ID (IPv4 address) of the neighbor’s elected DR and BDR.
Last Received Db Desc	The content of the last database description received from the specified neighbor.
Number of LSAs in Db Desc retransmitting	The number of LSAs that need to be retransmitted to the specified neighbor.
Number of LSAs in Summary List	The number of LSAs in the neighbor’s summary list.
Number of LSAs in Request List	The number of LSAs in the neighbor’s request list.
Number of LSAs in Retransmit List	The number of LSAs in the neighbor’s retransmit list.
Seqnum Mismatch	The number of times sequence number mismatches occurred.

TABLE 325 Detailed OSPFv3 neighbor information (Continued)

Field	Description
BadLSReq	The number of times the neighbor received a bad link-state request from the device.
One way received	The number of times a hello packet, which does not mention the router, is received from the neighbor. This omission in the hello packet indicates that the communication with the neighbor is not bidirectional.
Inactivity Timer	The number of times that the neighbor's inactivity timer expired.
Db Desc Retransmission	The number of times sequence number mismatches occurred.
LSReqRetrans	The number of times the neighbor retransmitted link-state requests to the device.
LSUpdateRetrans	The number of times the neighbor retransmitted link-state updates to the device.
LSA Received	The number of times the neighbor received LSAs from the device.
LS Update Received	The number of times the neighbor received link-state updates from the device.

Displaying routes redistributed into OSPFv3

You can display all IPv6 routes or a specified IPv6 route that the device has redistributed into OSPFv3.

To display all IPv6 routes that the device has redistributed into OSPFv3, enter the following command at any level of the CLI.

```
NetIron# show ipv6 ospf redistribute route
Id     Prefix                                    Protocol  Metric Type  Metric
 snIpAsPathAccessListStringRegExpression
1      2002::/16                                 Static    Type-2       1
2      2002:1234::/32                            Static    Type-2       1
```

Syntax: show ipv6 ospf redistribute route [<*ipv6-prefix*>]

The <*ipv6-prefix*> parameter specifies an IPv6 network prefix. (You do not need to specify the length of the prefix.)

For example, to display redistribution information for the prefix 2002::, enter the following command at any level of the CLI.

```
NetIron# show ipv6 ospf redistribute route 2002::
Id     Prefix                                    Protocol  Metric Type  Metric
1      2002::/16                                 Static    Type-2       1
```

These displays show the following information.

TABLE 326 OSPFv3 redistribution information

This field...	Displays...
ID	An ID for the redistributed route.
Prefix	The IPv6 routes redistributed into OSPFv3.

TABLE 326 OSPFv3 redistribution information (Continued)

This field...	Displays...
Protocol	The protocol from which the route is redistributed into OSPFv3. Redistributed protocols can be the following: • BGP – BGP4+. • RIP – RIPng. • ISIS – IPv6 IS-IS. • Static – IPv6 static route table. • Connected – A directly connected network.
Metric Type	The metric type used for routes redistributed into OSPFv3. The metric type can be the following: • Type-1 – Specifies a small metric (2 bytes). • Type-2 – Specifies a big metric (3 bytes).
Metric	The value of the default redistribution metric, which is the OSPF cost of redistributing the route into OSPFv3.

Displaying OSPFv3 route information

You can display the entire OSPFv3 route table for the device or only the route entries for a specified destination.

To display the entire OSPFv3 route table for the device, enter the following command at any level of the CLI.

```
NetIron#sh ipv6 ospf route
  Current Route count: 4
    Intra: 4 Inter: 0 External: 0 (Type1 0/Type2 0)
    Equal-cost multi-path: 0
    OSPF Type: IA- Intra, OA - Inter, E1 - External Type1, E2 - External Type2
   Destination                    Cost       E2Cost     Tag        Flags    Dis
IA 3000:200:1::1/128              0          0          0          00000003 110
   Next_Hop_Router                Outgoing_Interface Adv_Router
   ::                             loopback 1         10.1.2.1
   Destination                    Cost       E2Cost     Tag        Flags    Dis
IA 4000:200:1::1/128              0          0          0          00000003 110
   Next_Hop_Router                Outgoing_Interface Adv_Router
   ::                             loopback 2         10.1.2.1
   Destination                    Cost       E2Cost     Tag        Flags    Dis
IA 3000:300:1::1/128              0          0          0          00000003 110
   Next_Hop_Router                Outgoing_Interface Adv_Router
   ::                             loopback 1         10.1.2.1
   Destination                    Cost       E2Cost     Tag        Flags    Dis
IA 4000:300:1::1/128              0          0          0          00000003 110
   Next_Hop_Router                Outgoing_Interface Adv_Router
   ::                             loopback 2         10.1.2.1
```

Syntax: show ipv6 ospf routes [<*ipv6-prefix*>]

The <*ipv6-prefix*> parameter specifies a destination IPv6 prefix. (You do not need to specify the length of the prefix.) If you use this parameter, only the route entries for this destination are shown.

For example, to display route information for the destination prefix 2000:4::, enter the following command at any level of the CLI.

```
 NetIron# show ipv6 ospf route 2000::
Destination                 Cost       E2Cost     Tag        Flags      Dis
IA 2000::/64                   1          0         0         00000003   110
   Next_Hop_Router              Outgoing_Interface Adv_Router
   ::                              eth 1/1             10.1.1.1
```

These displays show the following information.

TABLE 327 OSPFv3 route information

This field...	Displays...
Current Route Count (Displays with the entire OSPFv3 route table only)	The number of route entries currently in the OSPFv3 route table.
Intra/Inter/External (Type1/Type2) (Displays with the entire OSPFv3 route table only)	The breakdown of the current route entries into the following route types: • Inter – The number of routes that pass into another area. • Intra – The number of routes that are within the local area. • External1 – The number of type 1 external routes. • External2 – The number of type 2 external routes.
Equal-cost multi-path (Displays with the entire OSPFv3 route table only)	The number of equal-cost routes to the same destination in the OSPFv3 route table. If load sharing is enabled, the router equally distributes traffic among the routes.
Destination	The IPv6 prefixes of destination networks to which the device can forward IPv6 packets. "*IA" indicates the next router is an intra-area router.
Cost	The type 1 cost of this route.
E2 Cost	The type 2 cost of this route.
Tag	The route tag for this route.
Flags	Flags associated with this route.
Dis	Administrative Distance for this route.
Next-Hop Router	The IPv6 address of the next router a packet must traverse to reach a destination.
Outgoing Interface	The router interface through which a packet must traverse to reach the next-hop router.
Adv_Router	The IP address of the advertising router.

Displaying OSPFv3 SPF information

You can display the following OSPFv3 SPF information:

- SPF node information for a specified area.
- SPF table for a specified area.
- SPF tree for a specified area.

For example, to display information about SPF nodes in area 0, enter the following command at any level of the CLI.

```
NetIron# show ipv6 ospf spf node area 0
SPF node for Area 0
SPF node 223.223.223.223,  cost: 0,  hops: 0
 nexthops to node:
 parent nodes:
 child nodes: 223.223.223.223:88

SPF node 223.223.223.223:88,  cost: 1,  hops: 1
 nexthops to node:    :: ethe 3/2
 parent nodes: 223.223.223.223
 child nodes: 1.1.1.1:0

SPF node 1.1.1.1:0,  cost: 1,  hops: 2
 nexthops to node:    fe80::2e0:52ff:fe91:bb37 ethe 3/2
 parent nodes: 223.223.223.223:88
 child nodes:
```

Syntax: show ipv6 ospf spf node area [<*area-id*>]

The **node** keyword displays SPF node information.

The **area** <*area-id*> parameter specifies a particular area. You can specify the <*area-id*> in the following formats:

- As an IPv4 address; for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

This display shows the following information.

TABLE 328 OSPFv3 SPF node information

This field...	Displays...
SPF node	Each SPF node is identified by its router ID (IPv4 address). If the node is a child node, it is additionally identified by an interface on which the node can be reached appended to the router ID in the format <*router-id*>:<*interface-id*>.
Cost	The cost of traversing the SPF node to reach the destination.
Hops	The number of hops needed to reach the parent SPF node.
Next Hops to Node	The IPv6 address of the next hop-router or the router interface through which to access the next-hop router.
Parent Nodes	The SPF node's parent nodes. A ***parent node*** is an SPF node at the highest level of the SPF tree, which is identified by its router ID.
Child Nodes	The SPF node's child nodes. A ***child node*** is an SPF node at a lower level of the SPF tree, which is identified by its router ID and interface on which the node can be reached.

For example, to display the SPF table for area 0, enter the following command at any level of the CLI.

```
NetIron# show ipv6 ospf spf table area 0
  SPF table for Area 0
  Destination         Bits Options  Cost  Nexthop                   Interface
R 1.1.1.1             ---- V6E---R-    1  fe80::2e0:52ff:fe91:bb37  ethe 3/2
N 223.223.223.223[88] ---- V6E---R-    1  ::                        ethe 3/2
```

Syntax: show ipv6 ospf spf table area <*area-id*>

The **table** parameter displays the SPF table.

The **area** <*area-id*> parameter specifies a particular area. You can specify the <*area-id*> in the following formats:

- As an IPv4 address, for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

This display shows the following information.

TABLE 329 OSPFv3 SPF table

This field...	Displays...
Destination	The destination of a route, which is identified by the following: • "R", which indicates the destination is a router. "N", which indicates the destination is a network. • An SPF node's router ID (IPv4 address). If the node is a child node, it is additionally identified by an interface on which the node can be reached appended to the router ID in the format <*router-id*>:<*interface-id*>.
Bits	A bit that indicates the capability of the device. The bit can be set to one of the following: • B – The device is an area border router. • E – The device is an AS boundary router. • V – The device is a virtual link endpoint. • W – The device is a wildcard multicast receiver.
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The router should be included in IPv6 routing calculations. E – The router floods AS-external-LSAs as described in RFC 2740. MC – The router forwards multicast packets as described in RFC 1586. N – The router handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The router handles demand circuits.
Cost	The cost of traversing the SPF node to reach the destination.
Next hop	The IPv6 address of the next hop-router.
Interface	The router interface through which to access the next-hop router.

For example, to display the SPF tree for area 0, enter the following command at any level of the CLI.

```
NetIron# show ipv6 ospf spf tree area 0
   SPF tree for Area 0
 +- 223.223.223.223 cost 0
     +- 223.223.223.223:88 cost 1
         +- 1.1.1.1:0 cost 1
```

Syntax: show ipv6 ospf spf tree area <*area-id*>

The **tree** keyword displays the SPF table.

The **area** <*area-id*> parameter specifies a particular area. You can specify the <*area-id*> in the following formats:

- As an IPv4 address; for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

In this sample output, consider the SPF node with the router ID 223.223.223.223 to be the top (root) of the tree and the local router. Consider all other layers of the tree (223.223.223.223:88 and 1.1.1.1:0) to be destinations in the network. Therefore, traffic destined from router 223.223.223.223 to router 1.1.1.1:0 must first traverse router 223.223.223.223:88.

Displaying IPv6 OSPF virtual link information

To display OSPFv3 virtual link information on a device, enter the following command.

```
NetIron# show ipv6 ospf virtual-link
Index Transit Area ID  Router ID        Interface Address             State
1     1                1.1.1.1          3003::2                       P2P
```

Syntax: show ipv6 ospf virtual-link

This display shows the following information.

TABLE 330 OSPFv3 virtual link information

This field...	Displays...
Index	An index number associated with the virtual link.
Transit Area ID	The ID of the shared area of two ABRs that serves as a connection point between the two routers.
Router ID	Router ID of the router at the other end of the virtual link (virtual neighbor).
Interface Address	The local address used to communicate with the virtual neighbor.
State	The state of the virtual link. Possible states include the following: • P2P – The link is functioning as a point-to-point interface. • DOWN – The link is down.

Displaying OSPFv3 virtual neighbor information

To display OSPFv3 virtual neighbor information for the device, enter the following command at the enabled level of the CLI.

```
NetIron#show ipv6 ospf virtual-neighbor
Index Router ID        Address                       State     Interface
1     14.14.14.14      2004:44:44:44::4              Full      eth 1/8
                Option: 00-00-00    QCount: 0    Timer: 408
2     14.14.14.14      2004:44:44:44::4              Full      tunnel 256
                Option: 00-00-00    QCount: 0    Timer: 43
```

Syntax: show ipv6 ospf virtual-neighbor [brief]

The [brief] option results in an output that omits the Option, QCount, and Timer fields. The command output shows the following information.

TABLE 331 OSPFv3 virtual neighbor information

This field...	Displays...
Index	An index number associated with the virtual neighbor.
Router ID	IPv4 address of the virtual neighbor.
Address	The IPv6 address to be used for communication with the virtual neighbor.

TABLE 331 OSPFv3 virtual neighbor information (Continued)

This field...	Displays...
State	The state between the device and the virtual neighbor. The state can be one of the following: • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
Interface	The IPv6 address of the virtual neighbor.
Option	The bits set in the virtual-link hello or database descriptors.
QCount	The number of packets that are in the queue and ready for transmission. If the system is stable, this number should always be 0.
Timer	A timer that counts down until a hello packet should arrive. If "timers" elapses and a hello packet has not arrived, the VL neighbor is declared to be down.

IPsec examples

This section contains examples of IPsec configuration and the output from the IPsec-specific **show** commands. In addition, IPsec-related information appears in general **show** command output for interfaces and areas.

The **show** commands that are specific to IPsec are:

- **show ipsec sa**
- **show ipsec policy**
- **show ipsec statistics**

The other **show** commands with IPsec-related information are:

- **show ipv6 ospf area**
- **show ipv6 ospf interface**
- **show ipv6 ospf vrf**

Showing IPsec security association information

The **show ipsec sa** command displays the IPSec security association databases, as follows.

```
NetIron#show ipsec sa
IPSEC Security Association Database(Entries:8)
SPDID(vrf:if) Dir Encap SPI        Destination          AuthAlg  EncryptAlg
1:ALL         in  ESP   512        35:1:1::1            sha1     Null
1:e1/1        out ESP   302        ::                   sha1     Null
1:e1/1        in  ESP   302        FE80::               sha1     Null
1:e1/1        out ESP   512        10:1:1::2            sha1     Null
2:ALL         in  ESP   512        35:1:1::1            sha1     Null
2:e1/2        out ESP   302        ::                   sha1     Null
2:e1/2        in  ESP   302        FE80::               sha1     Null
2:e1/2        out ESP   512        10:1:1::2            sha1     Nul
```

Syntax: show ipsec sa

Showing IPsec policy

The **show ipsec policy** command displays the database for the IPSec security policies. The fields for this **show** command output appear in the screen output example that follows. However, you should understand the layout and column headings for the display before trying to interpret the information in the example screen.

Each policy entry consists of two categories of information:

- The policy information
- The SA used by the policy

The policy information line in the screen begins with the heading Ptype and also has the headings Dir, Proto, Source (Prefix:TCP.UDP Port), and Destination (Prefix:TCP/UDPPort). The SA line contains the SPDID, direction, encapsulation (always ESP in the current release), the user-specified SPI, For readability, the policy information is described in Table 332, and SA-specific information is in Table 333.

```
NetIron#show ipsec policy
            IPSEC Security Policy Database(Entries:8)
PType  Dir Proto Source(Prefix:TCP/UDP Port)    Destination(Prefix:TCP/UDPPort)
SA: SPDID(vrf:if) Dir Encap SPI       Destination
use    in  OSPF  FE80::/10:any                  ::/0:any
SA: 2:e1/2        in  ESP   302        FE80::
use    out OSPF  FE80::/10:any                  ::/0:any
SA: 2:e1/2        out ESP   302        ::
use    in  OSPF  FE80::/10:any                  ::/0:any
SA: 1:e1/1        in  ESP   302        FE80::
use    out OSPF  FE80::/10:any                  ::/0:any
SA: 1:e1/1        out ESP   302        ::
use    in  OSPF  35:1:1::1/128:any              10:1:1::2/128:any
SA: 1:ALL         in  ESP   512        10:1:1::2
use    out OSPF  10:1:1::2/128:any              35:1:1::1/128:any
SA: 1:e1/1        out ESP   512        35:1:1::1
use    in  OSPF  35:1:1::1/128:any              10:1:1::2/128:any
SA: 2:ALL         in  ESP   512        10:1:1::2
use    out OSPF  10:1:1::2/128:any              35:1:1::1/128:any
```

Syntax: show ipsec policy

This command takes no parameters.

TABLE 332 IPsec policy information

This field...	Displays...
PType	This field contains the policy type. Of the existing policy types, only the “use” policy type is supported, so each entry can have only “use.”
Dir	The direction of traffic flow to which the IPsec policy is applied. Each direction has its own entry.
Proto	The only possible routing protocol for the security policy in the current release is OSPFv3.

TABLE 332 IPsec policy information (Continued)

This field...	Displays...
Source	The source address consists of the IPv6 prefix and the TCP or UDP port identifier.
Destination	The destination address consists of the IPv6 prefix. Certain logical elements have a bearing on the meaning of the destination address and its format, as follows: For IPsec on an interface or area, the destination address is shown as a prefix of 0xFE80 (link local). The solitary "::" (no prefix) indicates a "do not-care" situation because the connection is multicast. In this case, the security policy is enforced without regard for the destination address. For a virtual link (SPDID = 0), the address is required.

TABLE 333 SA used by the policy

This field...	Displays...
SA	This heading points at the SA-related headings for information used by the security policy. Thereafter, on each line of this part of the IPsec entry (which alternates with lines of policy information Table 332), "SA:" points at the fields under those SA-related headings. The remainder of this table describes each of the SA-related items.
SPDID	The security policy database identifier (SPDID) consists of two parts; the first part is an VRF id and the second part is an interface ID. The SPDID 0/ALL is a global database for the default VRF that applies to all interfaces.
Dir	The Dir field is either 'in" for inbound or "out" for outbound.
Encap	The type of encapsulation in the current release is ESP.
SPI	Security parameter index.
Destination	The IPv6 address of the destination endpoint. From the standpoint of the near interface and the area, the destination is not relevant and therefore appears as ::/0:any. For a virtual link, both the inbound and outbound destination addresses are relevant.

Showing IPsec statistics

The **show ipsec statistics** command displays the error and other counters for IPSec, as this example shows.

```
NetIron#show ipsec statistics
                        IPSecurity Statistics
secEspCurrentInboundSAs 1          ipsecEspTotalInboundSAs:  2
secEspCurrentOutboundSA 1          ipsecEspTotalOutboundSAs: 2
                    IPSecurity Packet Statistics
secEspTotalInPkts:      19         ipsecEspTotalInPktsDrop:  0
secEspTotalOutPkts:     83
                    IPSecurity Error Statistics
secAuthenticationErrors 0
secReplayErrors:        0          ipsecPolicyErrors:        13
secOtherReceiveErrors:  0          ipsecSendErrors:          0
secAuthenticationErrors 0
secReplayErrors:        0          ipsecPolicyErrors:        13
secOtherReceiveErrors:  0          ipsecSendErrors:          0
secUnknownSpiErrors:    0
```

Syntax: show ipsec statistics

This command takes no parameters.

Displaying IPsec configuration for an area

The **show ipv6 ospf area** [<*area-id*>] command includes information about IPsec for one area or all areas. In the example that follows, the IPsec information is in bold. IPsec is enabled in the first area (area 0) in this example but not in area 3. Note that in area 3, the IPsec key was specified as not encrypted.

```
NetIron(config-ospf6-router)#show ipv6 ospf area
  Authentication: Configured
   KeyRolloverTime(sec): Configured: 25 Current: 20
   KeyRolloverState: Active,Phase1
   Current: None
   New: SPI:400, ESP, SHA1
      Key:$Z|83OmYW{QZ|83OmYW{QZ|83OmYW{QZ|83OmYW{Q
  Interface attached to this area: eth 1/1
  Number of Area scoped LSAs is 6
  Sum of Area LSAs Checksum is 0004f7de
  Statistics of Area 0:
    SPF algorithm executed 6 times
    SPF last updated: 482 sec ago
    Current SPF node count: 1
      Router: 1 Network: 0
      Maximum of Hop count to nodes: 0
Area 3:
  Authentication: Not Configured
  Interface attached to this area:
  Number of Area scoped LSAs is 3
```

Syntax: show ipv6 ospf area [<*area-id*>]

The <*area-id*> parameter restricts the display to the specified OSPF area. You can specify the <*area-id*> parameter in the following formats:

- An IPv4 address, for example, 192.168.1.1
- A numerical value in the range 0 – 2,147,483,647

TABLE 334 Area configuration of IPsec

This field...	Displays...
Authentication	This field shows whether or not authentication is configured. If this field says "Not Configured," the IPsec-related fields (bold in example screen output) are not displayed at all.
KeyRolloverTime	The number of seconds between each initiation of a key rollover. This field shows the configured and current times.
KeyRolloverState	Can be: Not active: key rollover is not active> Active phase 1: rollover is in its first interval. Active phase 2: rollover is in its second interval.
Current	Shows current SPI, authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the current key.
New	Shows new SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the new key.
Old	Shows old SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the old key.

Displaying IPsec for an interface

To see IPsec configuration for a particular interface or all interfaces, use the **show ipv6 ospf interface** command as in the following example (IPsec information in bold).

```
NetIron#show ipv6 ospf interface
eth 1/3 is down, type BROADCAST
   Interface is disabled

eth 1/8 is up, type BROADCAST
  IPv6 Address:
      2100:18:18:18:18::1/64
      2100:18:18:18::/64
  Instance ID 255, Router ID 1.1.1.1
  Area ID 1, Cost 1
  State BDR, Transmit Delay 1 sec, Priority 1
  Timer intervals :
    Hello 10, Hello Jitter 10  Dead 40, Retransmit 5
  Authentication: Enabled
   KeyRolloverTime(sec): Configured: 30 Current: 0
   KeyRolloverState: NotActive
   Outbound: SPI:121212, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
   Inbound: SPI:121212, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
  DR:2.2.2.2 BDR:1.1.1.1  Number of I/F scoped LSAs is 2
  DRElection:     1 times, DelayedLSAck:    83 times
  Neighbor Count = 1,   Adjacent Neighbor Count= 1
     Neighbor:
      2.2.2.2 (DR)
    Statistics of interface eth 1/8:
      Type     tx         rx         tx-byte    rx-byte
      Unknown  0          0          0          0
      Hello    1415       1408       56592      56320
      DbDesc   3          3          804        804
      LSReq    1          1          28         28
      LSUpdate 193        121        15616      9720
      LSAck    85         109        4840       4924
      OSPF messages dropped,no authentication: 0
```

Syntax: show ipv6 ospf interface [ethernet <*slot/port*> | pos <*slot/port*> | loopback <*number*> | tunnel <*number*> | ve <*number*>]

TABLE 335 Area configuration of IPsec

This field...	Displays...
Authentication	This field shows whether or not authentication is configured. If this field says "Not Configured," the IPsec-related fields (bold in example screen output) are not displayed at all.
KeyRolloverTime	The number of seconds between each initiation of a key rollover. This field shows the configured and current times.
KeyRolloverState	Can be: Not active: key rollover is not active> Active phase 1: rollover is in its first interval. Active phase 2: rollover is in its second interval.
Current	Shows current SPI, authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the current key.
New (Inbound or Outbound)	Shows new SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the new key.

TABLE 335 Area configuration of IPsec (Continued)

This field...	Displays...
Old (Inbound or Outbound)	Shows old SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the old key.
OSPF messages dropped	Shows the number of packets dropped because the packets failed authentication (for any reason).

Displaying IPsec for a virtual link

To display IPsec for a virtual link, run the **show ipv6 ospf virtual-link brief** or **show ipv6 ospf virtual-link** command, as the following examples illustrate.

```
NetIron#show ipv6 ospf virtual-link brief
Index Transit Area ID  Router ID        Interface Address             State
1     1                14.14.14.14      3000:1:1:1::1                 P2P

NetIron#show ipv6 ospf virtual-link
Transit Area ID  Router ID        Interface Address             State
1                14.14.14.14      3000:1:1:1::1                 P2P
  Timer intervals(sec) :
    Hello 10, Hello Jitter 10,  Dead 40, Retransmit 5, TransmitDelay 1
  DelayedLSAck:     5 times
  Authentication: Configured
   KeyRolloverTime(sec): Configured: 10 Current: 0
   KeyRolloverState: NotActive
   Outbound: SPI:100004, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
   Inbound: SPI:100004, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
  Statistics:
      Type     tx         rx         tx-byte    rx-byte
      Unknown  0          0          0          0
      Hello    65         65         2600       2596
      DbDesc   4          4          2752       2992
      LSReq    1          1          232        64
      LSUpdate 11         5          1040       1112
      LSAck    5          8          560        448
      OSPF messages dropped,no authentication: 0
Neighbor: State: Full Address: 2004:44:44:44::4 Interface: eth 2/2
```

Syntax: show ipv6 ospf virtual-link [brief]

The optional [brief] keyword limits the display to the Transit, Area ID, Router ID, Interface Address, and State fields for each link.

Changing a key

In this example, the key is changed as illustrated in the two command lines that follow. Note that the SPI value is changed from 300 to 310 to comply with the requirement that you change the SPI when you change the key.

Initial configuration command.

```
NetIron(config-if-e10000-1/3)#ipv6 ospf auth ipsec spi 300 esp sha1
no-encrypt 1234567890098765543123456789 0aabbccddef
```

Command line for changing the key.

```
NetIron(config-if-e10000-1/3)#ipv6 ospf auth ipsec spi 310 esp sha1
no-encrypt 9898989890098765543212345678 90aabbccddef
```

Displaying IPv6 OSPF information for a VRF

To display IPv6 OSPF information for a VRF or all VRF interfaces, use the **show ipv6 ospf vrf** command as in the following example.

```
PowerConnect#show ipv6 ospf vrf red
OSPFv3 Process number 0 with Router ID 0x02020202(2.2.2.2)
Running 0 days 0 hours 5 minutes 49 seconds
Number of AS scoped LSAs is 0
Sum of AS scoped LSAs Checksum is 00000000
External LSA Limit is 250000
Database Overflow Interval is 10
Database Overflow State is NOT OVERFLOWED
Route calculation executed 0 times
Pending outgoing LSA count 0
Authentication key rollover interval 30 seconds
Number of areas in this router is 4
Router is operating as ABR
Router is operating as ASBR, Redistribute: CONNECTED
High Priority Message Queue Full count: 0
BFD is disabled
```

Syntax: show ipv6 ospf vrf <vrf-name> **area** [<area-id>] | [<virtual-links>]

The <vrf-name> parameter specifies the VRF that you want the OSPF area information for.

The <area-id> parameter shows information for the specified area.

The <virtual-link> parameter displays the entry that corresponds to the IP address you enter.

Use the **show ipv6 ospf vrf** command to display the currently selected IPv6 global address for use by the Virtual Links in each transit area.

```
PowerConnect#show ipv6 ospf vrf red area
Area 3:
        Authentication: Not Configured
        Interface attached to this area:
        Number of Area scoped LSAs is 3
        Sum of Area LSAs Checksum is 0001a6c4
        Statistics of Area 3:
                SPF algorithm executed 3 times
                SPF last updated: 302 sec ago
                Current SPF node count: 1
                Router: 1 Network: 0
                Maximum of Hop count to nodes: 0
Area 2:
        Authentication: Not Configured
        Interface attached to this area:
        Number of Area scoped LSAs is 3
        Sum of Area LSAs Checksum is 000192d6
        Statistics of Area 2:
                SPF algorithm executed 3 times
                SPF last updated: 302 sec ago
                Current SPF node count: 1
                Router: 1 Network: 0
                Maximum of Hop count to nodes: 0
Area 1:
        Authentication: Not Configured
        Interface attached to this area: eth 1/1
        Number of Area scoped LSAs is 6
        Sum of Area LSAs Checksum is 00046630
        Statistics of Area 1:
                SPF algorithm executed 3 times
                SPF last updated: 302 sec ago
                Current SPF node count: 3
                Router: 2 Network: 1
                Maximum of Hop count to nodes: 2
                Global IPv6 Address used by Virtual Links in this area:10:1:1::2
Area 0.0.0.0 :
        Authentication: Not Configured
        Interface attached to this area: VLink 1
        Number of Area scoped LSAs is 6
        Sum of Area LSAs Checksum is 0002cc53
        Statistics of Area 0.0.0.0:
                SPF algorithm executed 3 times
                SPF last updated: 302 sec ago
                Current SPF node count: 2
```

Syntax: show ipv6 ospf vrf <vrf-name> **area** [<area-id>] | [<virtual-links>]

Use the **show ipv6 ospf vrf** neighbor command to display the currently selected neighbor for use by the Virtual Links in each transit area.

```
NetIron#sh ipv6 ospf vrf red neighbor
Total number of neighbors in all states: 1
Number of neighbors in state Full      : 1

Type     tx         rx         tx-byte    rx-byte
      Unknown  0          0          0          0
      Hello    32         32         1276       1280
      DbDesc   2          2          116        116
      LSReq    1          1          52         52
      LSUpdate 2          2          184        200
      LSAck    2          2          112        112
      OSPF messages dropped,no authentication: 0
  Neighbor: State: Full Address: 35:1:1::1 Interface: eth 1/1
```

OSPFv3 clear commands

The following OSPFv3 clear commands are supported.

Clearing all OSPFv3 data

You can use the **ospf all** command to clear all OSPF data by disabling and enabling the OSPFv3 processes as shown in the following.

```
NetIron# clear ipv6 ospf all
```

Syntax: clear ipv6 ospf all

Clearing all OSPFv3 packet counters

You can use the **ospf traffic** command to clear all OSPFv3 packet counters as shown in the following.

```
NetIron# clear ipv6 ospf traffic
```

Syntax: clear ipv6 ospf traffic

Scheduling Shortest Path First (SPF) calculation

You can use the **ospf force-spf** command to perform the SPF calculation without clearing the OSPF database, as shown in the following.

```
NetIron# clear ipv6 ospf force-spf
```

Syntax: clear ipv6 ospf force-spf

Clearing all redistributed routes from OSPF

You can use the **ospf redistribution** command to clear all redistributed routes from OSPF, as shown in the following.

```
NetIron# clear ipv6 ospf redistribution
```

Syntax: clear ipv6 ospf redistribution

Clearing OSPF neighbors

You can use the **ospf neighbor** command to delete and relearn OSPF neighbors, as shown in the following:

- Clearing all OSPF Neighbors
- Clearing OSPF Neighbors Attached to a Specified Interface

Clearing all OSPF neighbors

You can use the **ospf neighbor all** command to delete and relearn all OSPF neighbors, as shown in the following.

```
NetIron# clear ipv6 ospf neighbor all
```

Syntax: clear isv6 ospf neighbor all

Clearing OSPF neighbors attached to a specified interface

You can use the **ospf neighbor interface** command to delete and relearn the OSPF neighbors attached to a specified interface, as shown in the following.

```
NetIron# clear ipv6 ospf neighbor interface ethernet 1/1
```

Syntax: clear isv6 ospf neighbor interface ethernet <*slot/port*> **| pos** <*slot/port*> **| ve** <*port-no*> **| tunnel** <*tunnel-port*> [<*nbrid*>]

Specify the interface options as shown in the following options.

ethernet <*slot/port*> – clears OSPF neighbors on the specified Ethernet interface.

pos <*slot/port*> – clears OSPF neighbors on the specified POS interface.

ve <*port-no*> – clears OSPF neighbors on the specified virtual interface.

tunnel <*tunnel-port*> – clears OSPF neighbors on the specified tunnel interface.

Specifying the <*nbrid*> variable limits the **clear ipv6 ospf neighbor** command to an individual OSPF neighbor attached to the interface.

Clearing OSPF counters

You can use the **ospf counts** command to clear OSPF neighbor's counters as described in the following:

- Clearing all OSPF Counters
- Clearing the OSPF Counters for a Specified Neighbor
- Clearing the OSPF Counters for a Specified Interface

Clearing all OSPF counters

You can clear all OSPF counters using the **clear ipv6 counts** command, as shown in the following.

```
NetIron# clear ipv6 ospf counts
```

Syntax: clear ipv6 ospf counts

Clearing OSPF counters for a specified neighbor

You can clear all OSPF counters for a specified neighbor using the **clear ipv6 counts neighbor** command, as shown in the following.

```
NetIron# clear ipv6 ospf counts neighbor 136.10.10.1
```

Syntax: clear ipv6 ospf counts neighbor <*nbrid*>

The <*nbrid*> variable specifies the neighbor ID of the OSPF neighbor whose counters you want to clear.

Clearing OSPF counters for a specified interface

You can clear all OSPF counters for a specified interface using the **clear ipv6 counts neighbor interface** command, as shown in the following.

```
NetIron# clear ipv6 ospf counts interface ethernet 3/1
```

Syntax: clear ipv6 ospf counts neighbor interface ethernet <*slot/port*> **| pos** <*slot/port*> **| ve** <*port-no*> **| tunnel** <*tunnel-port*> [<*nbrid*>]

Specify the interface options as shown in the following options.

ethernet <*slot/port*> – clears OSPF counters for OSPF neighbors on the specified Ethernet interface.

pos <*slot/port*> – clears OSPF counters for OSPF neighbors on the specified POS interface.

ve <*port-no*> – clears OSPF counters for OSPF neighbors on the specified virtual interface.

tunnel <*tunnel-port*> – clears OSPF counters for OSPF neighbors on the specified tunnel interface.

Using an <*nbrid*> value limits the displayed output to an individual OSPF neighbor attached to the interface.

Chapter

44

Configuring IPv6 IS-IS

PowerConnect B-MLXe supports the following IPv6 IS-IS features:

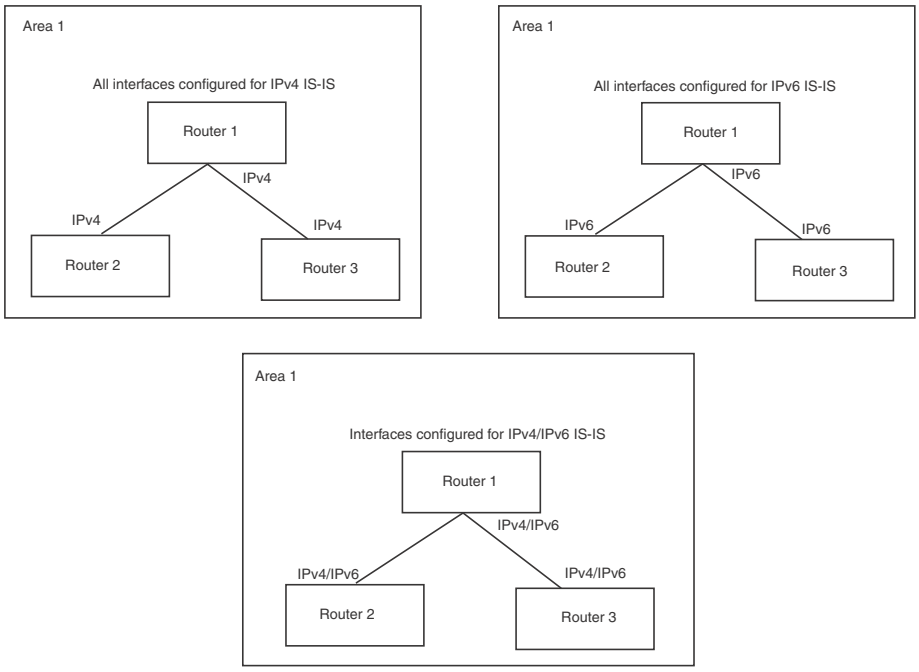
- IPv6 IS-IS
- Redistributing BGP4+ Routes into IPv6 IS-IS
- Redistributing RIPng Routes into IPv6 IS-IS
- Redistributing OSPFv3. Routes into IPv6 IS-IS
- Redistributing Static IPv6 Routes into IPv6 IS-IS
- Redistributing IPv6 routes learned from directly connected networks
- IPv6 Protocol-Support Consistency Checks

A description of the IS-IS protocol is provided in 25, “Configuring IS-IS (IPv4)”. This chapter describes the specific requirements for configuring a PowerConnect router for IPv6 IS-IS.

IPv6 IS-IS single-topology mode

IPv6 IS-IS supports single-topology mode, which means that you can run IPv6 IS-IS concurrently with other network protocols such as IPv4 IS-IS throughout a topology. However, when implementing a single topology, all routers in an area (Level 1 routing) or domain (Level 2 routing) must be configured with the same set of network protocols on all its interfaces, even on loopback interfaces. You can configure IPv4 IS-IS only, IPv6 IS-IS only, or both IPv4 IS-IS and IPv6 IS-IS ((Figure). For example, to successfully implement both IPv4 and IPv6 IS-IS in an area, you must configure both IPv4 and IPv6 IS-IS on all router interfaces in the area.

FIGURE 218



A single shortest path first (SPF) per level computes the IPv4 and IPv6 routes. The use of a single SPF indicates that both IPv4 and IPv6 IS-IS routing protocols must share a common network topology

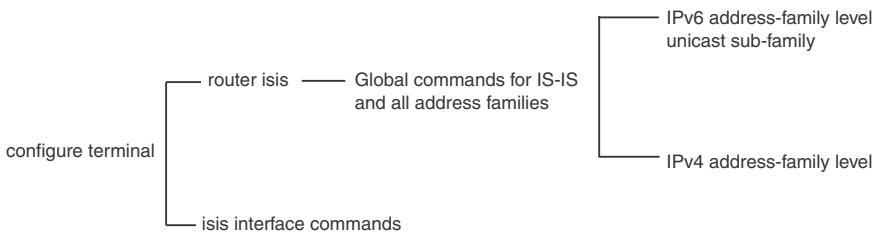
The implementation of IPv4 IS-IS supports type, length, and value (TLV) parameters to advertise reachability to IPv4 networks. The TLVs specify the types of data, the maximum length of the data, and the valid values for the data. IPv6 IS-IS advertises its information using new TLV parameters. The new TLV parameters for IPv6 support an extended default metric value.

In a single topology, if both IPv4 and IPv6 are configured on an interface, metric-style must be set to wide in both address families. Narrow is the default for IPv4. Wide is the default for IPv6.

IS-IS CLI levels

The CLI includes various levels of commands for IS-IS. Figure 219 diagrams these levels that includes the levels used for IPv6 IS-IS.

FIGURE 219 IPv6 IS-IS CLI levels



The IPv6 IS-IS CLI levels are as follows:

- A global level for the configuration of the IS-IS protocol. At this level, all IS-IS configurations at this level apply to IPv4 and IPv6. You enter this layer using the **router isis** command.
  - Under the global level, you specify an address family. Address families separate the IS-IS configuration IPv6 and IPv4. You enter configurations that are for a specific You enter this level by entering the **address-family** command at the router isis level.
  - Under the address family level, you select a sub-address family, which is the type of routes for the configuration. For IS-IS, you specify **unicast**.
- An interface level

Global configuration level

You enter the global configuration level of ISIS by entering the following command:

```
NetIron(config)#router isis
NetIron(config-isis-router)#
```

Syntax: router isis

The `(config-isis-router)#` prompt indicates that you are at the global level for IS-IS. Configurations you enter at this level apply to both IS-IS IPv4 and IS-IS IPv6.

Address family configuration level

The implementation of IPv6 IS-IS includes a new configuration level: address family. You enter IS-IS definitions for IPv6 IS-IS under this level. Address-family allows you to create configurations for IPv6 IS-IS unicast routes that are separate and distinct from configurations for IPv4 IS-IS unicast routes.

Under the address family level, Dell currently supports the unicast address family configuration level only. The device enters the IPv6 IS-IS unicast address family configuration level when you enter the following command while at the global IS-IS configuration level:

```
NetIron(config-isis-router)# address-family ipv6 unicast
NetIron(config-isis-router-ipv6u)#
```

Syntax: address-family ipv6 unicast

The `(config-isis-router-ipv6u)#` prompt indicates that you are at the IPv6 IS-IS unicast address family configuration level. While at this level, you can access several commands that allow you to configure IPv6 IS-IS unicast routes.

NOTE
Each address family configuration level allows you to access commands that apply to that particular address family only. To enable a feature in a particular address family, you must specify any associated commands for that feature in that particular address family. You cannot expect the feature, which you may have configured in the IPv4 IS-IS unicast address family, to work in the IPv6 IS-IS unicast address family unless it is explicitly configured in the IPv6 IS-IS unicast address family.

To exit from the IPv6 IS-IS unicast address family configuration level, enter the following command:

```
NetIron(config-isis-router-ipv6u)# exit-address-family
NetIron(config-isis-router)#
```

Entering this command returns you to the global IS-IS configuration level.

Interface level

Some IS-IS definitions are entered at the interface level. To change to the interface level for IS-IS configuration, enter the following command.

```
NetIron(config)# interface ethernet 2/3
NetIron(config-if-e1000-2/3)#ipv6 router isis
```

Syntax: ipv6 router isis

Configuring IPv6 IS-IS

Enabling IPv6 IS-IS globally

Follow the steps listed below to configure IPv6 IS-IS globally.

1. You must enable the forwarding of IPv6 traffic on the device using the **ipv6 unicast-routing** command. Enter a command such as the following:

   ```
   NetIron#configure terminal
   NetIron(config)# ipv6 unicast-routing
   ```

   Syntax: [no] ipv6 unicast-routing

2. Globally enable IS-IS by entering the following command:

   ```
   NetIron(config)# router isis
   ISIS: Please configure NET!
   ```

   Once you enter **router isis**, the device enters the IS-IS router configuration level.

   Syntax: [no] router isis

   To disable IS-IS, use the **no** form of this command.

3. If you have not already configured a NET for IS-IS, enter commands such as the following:

   ```
   NetIron(config-isis-router)# net 49.2211.aaaa.bbbb.cccc.00
   NetIron(config-isis-router)#
   ```

   The commands in the example above configure a NET that has the area ID 49.2211, the system ID aaaa.bbbb.cccc (the device's base MAC address), and SEL value 00.

   Syntax: [no] net <*area-id*>.<*system-id*>.<*sel*>

   The <*area-id*> parameter specifies the area and has the format xx or xx.xxxx. For example, 49 and 49.2211 are valid area IDs.

   The <*system-id*> parameter specifies the router's unique IS-IS router ID and has the format xxxx.xxxx.xxxx. You can specify any value for the system ID. A common practice is to use the device's base MAC address as the system ID. The base MAC address is also the MAC address of port 1. To determine the base MAC address, enter the following command at any level of the CLI: **show interfaces brief**. The base MAC address is listed in the first row of information, in the MAC column.

   You must use the same system ID in all the NETs on the device.

NOTE
The parameter descriptions above are the recommended values for the NET. However, the CLI accepts any value that fits within the following lengths and formats:

xx.xxxx.xxxx.xxxx.00 – minimum length of NET

xx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.00 – maximum length of NET

The <*sel*> parameter specifies the NSAP Selector (SEL). This value must always be 00 (two zeros). The value 00 indicates that this address is an NET.

To delete a NET, use the **no** form of this command.

4. Configure an IPv6 IS-IS single topology. Refer to "Configuring IPv6 IS-IS single topology" on page 1872.
5. Configure ISIS parameters. Refer to the sections "Globally configuring IS-IS on a device" on page 1872, "Configuring IPv6 specific address family route parameters" on page 1872, and "Configuring ISIS properties on an interface" on page 1879.

Enabling IS-IS and assigning an IPv6 address to an interface

To configure IPv6 IS-IS on the desired router interfaces, enter commands such as the following:

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e100-3/1)# ipv6 address 2001:200:12D:1300::/64 eui-64
NetIron(config-if-e100-3/1)# ipv6 router isis
```

The commands in this example assign the global IPv6 prefix 2001:200:12d:1300::/64 to Ethernet interface 3/1 and enable IPv6 IS-IS on the interface.

Syntax: ipv6 address <*ipv6-prefix*>/<*prefix-length*> **[eui-64]**

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **eui-64** keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Syntax: [no] ipv6 router isis

To disable IPv6 IS-IS on an interface, use the **no** form of this command.

The following configuration tasks are optional:

- Configure IPv6 route parameters.
- Redistribute routes from other route sources into IPv6 IS-IS.
- Perform IPv6 IS-IS adjacency checks.
- Disable partial SPF calculations

Configuring IPv6 IS-IS single topology

If your IS-IS single topology will support both IPv6 and IPv4, you can configure both IPv6 and IPv4 on an IS-IS interface for Level 1, Level 2, or both Level 1 and Level 2. However, if you configure both IPv6 and IPv4 on an IS-IS interface, they must be configured to run on the same level. For example, you can configure IPv6 to run on Level 1 on an interface and IPv4 to also run on Level 1 on the same interface. However, you cannot configure IPv6 to run on Level 1 on an interface and IPv4 to run to Level 2 on the same interface.

To configure an IPv6 IS-IS single topology, you must perform the tasks listed below.

1. Globally enable IS-IS and configure at least one Network Entity Title (NET). The NET is the device's network interface with IS-IS. You can configure up to three NETs on a device.
2. Configure the desired router interfaces with an IPv6 address and enable IPv6 IS-IS on the router interfaces.
3. Configure ISIS parameters. Refer to the sections "Globally configuring IS-IS on a device" on page 1872, "Configuring IPv6 specific address family route parameters" on page 1872, and "Configuring ISIS properties on an interface" on page 1879.

Globally configuring IS-IS on a device

The following configuration tasks described in 25, "Configuring IS-IS (IPv4)", apply to IS-IS IPv6 configuration:

- Setting the Overload Bit
- Configuring Authentication
- Changing the IS-IS Level Globally
- Disabling or Re-enabling Display of Hostname
- Changing the Sequence Numbers PDU Interval
- Changing the Maximum LSP Lifetime
- Changing the LSP Refresh Interval
- Changing the LSP Generation Interval
- Changing the LSP Interval and Retransmit
- Changing the SPF Timer
- Globally Disabling or Re-Enabling Hello Padding
- Logging Adjacency Changes
- Disabling Partial SPF Calculations.

Configuring IPv6 specific address family route parameters

This section describes how to modify the IS-IS the parameters for the IS-IS IPv6 address family.

Changing the maximum number of load sharing paths

By default, IPv6 IS-IS can calculate and install four equal-cost paths into the IPv6 forwarding table. You can change the number of paths IPv6 IS-IS can calculate and install in the IPv6 forwarding table to an amount from
1 – 8. If you change the number of paths to one, the device does not load share route paths learned from IPv6 IS-IS.

For example, to change the number of paths IPv6 IS-IS can calculate and install in the IPv6 forwarding table to three, enter the following command at the IPv6 IS-IS unicast address family configuration level:

```
NetIron(config-isis-router-ipv6u)# maximum-paths 8
```

Syntax: [no] maximum-paths <*number*>

The <*number*> parameter specifies the number of paths IPv6 IS-IS can calculate and install in the IPv6 forwarding table.

To return to the default number of maximum paths, enter the **no** form of this command.

Enabling advertisement of a default route

By default, the device does not generate or advertise a default route to its neighboring ISs. A default route is not advertised even if the device's IPv6 route table contains a default route. You can enable the device to advertise a default route to all neighboring ISs using one of the following methods. By default, the feature originates the default route at Level 2 only. However, you can apply a route map to originate the default route to Level 1 only or at both Level 1 and Level 2.

NOTE
This feature requires the presence of a default route in the IPv6 route table.

To enable the device to advertise a default route that is originated a Level 2, enter the following command at the IPv6 IS-IS unicast address family configuration level:

```
NetIron(config-isis-router-ipv6u)# default-information-originate
```

This command enables the device to advertise a default route into the IPv6 IS-IS area to which the device is attached.

Syntax: [no] default-information-originate [route-map <*name*>**]**

The **route-map** <*name*> parameter allows you to specify the level on which to advertise the default route. You can specify one of the following:

- Advertise to Level-1 ISs only.
- Advertise to Level-2 ISs only.
- Advertise to Level-1 and Level-2 ISs.

NOTE
The route map must be configured before you can use the route map as a parameter with the **default-information-originate** command.

To use a route map to specify the router to advertise a default route to Level 1, enter commands such as the following at the Global CONFIG level:

```
NetIron(config)# route-map default_level1 permit 1
NetIron(config-routemap default_level1)# set level level-1
NetIron(config-routemap default_level1)# router isis
NetIron(config-isis-router)# address-family ipv6 unicast
NetIron(config-isis-router-ipv6u)# default-information-originate route-map
default_level1
```

These commands configure a route map to set the default advertisement level to Level 1 only.

Syntax: [no] route-map <*map-name*> **permit | deny** <*sequence-number*>

Syntax: [no] set level level-1 | level-1-2 | level-2

For this use of a route map, use the **permit** option and do not specify a **match** statement. Specify a **set** statement to set the level to one of the following:

- **level-1** – Level 1 only.
- **level-1-2** – Level 1 and Level 2.
- **level-2** – Level 2 only (default).

Changing the administrative distance for IPv6 IS-IS

When the device has paths from multiple routing protocols to the same destination, it compares the administrative distances of the paths and selects the path with the lowest administrative distance to place in the IPv6 route table.

For example, if the router has a path from RIPng, from OSPFv3, and IPv6 IS-IS to the same destination, and all the paths are using their protocols' default administrative distances, the router selects the OSPFv3 path, because that path has a lower administrative distance than the RIPng and IPv6 IS-IS paths.

Here are the default IPv6 administrative distances on the device:

- Directly connected – 0 (this value is not configurable)
- Static – 1 (applies to all static routes, including default routes)
- EBGP – 20
- OSPFv3 – 110
- IPv6 IS-IS – 115
- RIPng – 120
- IBGP – 200
- Local BGP – 200
- Unknown – 255 (the device will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the device receives routes for the same network from IPv6 IS-IS and from RIPng, it will prefer the IPv6 IS-IS route by default.

To change the administrative distance for IPv6 IS-IS routes, enter the following command at the IPv6 IS-IS unicast address family configuration level:

```
NetIron(config-isis-router-ipv6u)# distance 100
```

Syntax: [no] distance <*number*>

This command changes the administrative distance for all IPv6 IS-IS routes to 100.

The <*number*> parameter specifies the administrative distance. You can specify a value from 1 – 255. (Routes with a distance value of 255 are not installed in the routing table.) The default for IPv6 IS-IS is 115.

Configuring summary prefixes

You can configure summary prefixes to aggregate IPv6 IS-IS route information. Summary prefixes can enhance performance by reducing the size of the Link State database, reducing the amount of data a router needs to send to its neighbors, and reducing the CPU cycles used for IPv6 IS-IS.

When you configure a summary prefix, the prefix applies only to Level-2 routes by default. You can specify Level-1 only, Level-2 only, or Level-1 and Level-2 when you configure the prefix.

For example, to configure a summary prefix of 2001:e0ff::/32 to be advertised to Level-1 routes only, enter the following command at the IPv6 IS-IS unicast address family configuration level:

```
NetIron(config-isis-router-ipv6u)# summary-prefix 2001:e0ff::/32 level-1
```

Syntax: [no] summary-prefix <*ipv6-prefix*>**/**<*prefix-length*> **[level-1 | level-1-2 | level-2-only]**

The <*ipv6-prefix*>/<*prefix-length*> parameter specifies the aggregate address. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **level-1** | **level-1-2** | **level-2-only** parameter specifies the route types to which the aggregate route applies. The default is **level-2-only**.

Redistributing routes into IPv6 IS-IS

To redistribute routes into IPv6 IS-IS, you can perform the following configuration tasks:

- Change the default redistribution metric (optional).
- Configure the redistribution of a particular route type into IPv6 IS-IS (mandatory).

The device can redistribute routes from the following route sources into IPv6 IS-IS:

- BGP4+.
- RIPng.
- OSPFv3.
- Static IPv6 routes.
- IPv6 routes learned from directly connected networks.

The device can also can redistribute Level-1 IPv6 IS-IS routes into Level-2 IPv6 IS-IS routes, and Level-2 IPv6 IS-IS routes into Level-1 IPv6 IS-IS routes.

Route redistribution from other sources into IPv6 IS-IS is disabled by default. When you enable redistribution, the device redistributes routes only into Level 2 by default. You can specify Level 1 only, Level 2 only, or Level 1 and Level 2 when you enable redistribution.

The device automatically redistributes Level-1 routes into Level-2 routes. Thus, you do not need to enable this type of redistribution. You also can enable redistribution of Level-2 routes into Level-1 routes.

The device attempts to use the redistributed route's metric as the route's IPv6 IS-IS metric. For example, if an OSPFv3 route has an OSPF cost of 20, the router uses 20 as the route's IPv6 IS-IS metric. The device uses the redistributed route's metric as the IPv6 IS-IS metric unless the route does not a have a valid metric. In this case, the device assigns the default metric value to the route. For information about the default metric, refer to the "Changing the Default Redistribution Metric" section, which follows this section.

Changing the default redistribution metric

When IPv6 IS-IS redistributes a route from another route source (such as OSPFv3, BGP4+, or a static IPv6 route) into IPv6 IS-IS, it uses the route's metric value as its metric when the metric is not modified by a route map or metric parameter and the default redistribution metric is set to its default value of 0. You can change the default metric to a value from 0 – 65535.

NOTE
The implementation of IS-IS does not support the optional metric types Delay, Expense, or Error.

For example, to change the default metric to 20, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# default-metric 20
```

Syntax: [no] **default-metric** <*number*>

The <*number*> parameter specifies the default metric. You can specify a value from 0 – 65535. The default is 0.

The restore the default value for the default metric, enter the **no** form of this command.

Redistributing static IPv6 routes into IPv6 IS-IS

To redistribute static IPv6 routes from the IPv6 static route table into IPv6 IS-IS routes, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# redistribute static
```

This command configures the device to redistribute all static IPv6 routes into Level-2 IS-IS routes.

Syntax: [no] **redistribute static [level-1 | level-1-2 | level-2 | metric** <*number*> **| metric-type external | internal | route-map** <*name*>**]**

The **level-1**, **level-1-2**, and **level-2** keywords restrict redistribution to the specified IPv6 IS-IS level.

The **metric** <*num*> parameter changes the metric. You can specify a value from 0 - 4294967295.

The **metric-type external** | **internal** parameter restricts redistribution to one of the following:

- **external** – The metric value is not comparable to an IPv6 IS-IS internal metric and is always higher than the IPv6 IS-IS internal metric.
- **internal** – The metric value is comparable to metric values used by IPv6 IS-IS. This is the default.

The **route-map** <*name*> parameter restricts redistribution to those routes that match the specified route map. The route map must already be configured before you use the route map name with the **redistribute** command. For example, to configure a route map that redistributes only the static IPv6 routes to the destination networks 2001:100::/32, enter commands such as the following.

```
NetIron(config)# ipv6 access-list static permit any 2001:100::/32
NetIron(config)# route-map static permit 1
NetIron(config-routemap static)# match ip address static
NetIron(config-routemap static)# router isis
NetIron(config-isis-router)# address-family ipv6 unicast
NetIron(config-isis-router-ipv6u)# redistribute static route-map static
```

For information about the IPv6 ACL and route map syntax, refer to the following:

- Chapter 40, “Configuring an IPv6 Access Control List”.

Redistributing directly connected routes into IPv6 IS-IS

To redistribute directly connected IPv6 routes into IPv6 IS-IS routes, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# redistribute connected
```

This command configures the device to redistribute all directly connected routes in the IPv6 route table into Level-2 IPv6 IS-IS.

Syntax: [no] **redistribute connected [level-1 | level-1-2 | level-2 | metric** <*number*> **| metric-type external | internal | route-map** <*name*>]

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing RIPng routes into IPv6 IS-IS

To redistribute RIPng routes into IPv6 IS-IS, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# redistribute rip
```

This command configures the device to redistribute all RIPng routes into Level-2 IS-IS.

Syntax: [no] **redistribute rip [level-1 | level-1-2 | level-2 | metric** <*number*> **| metric-type external | internal | route-map** <*name*>]

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing OSPF version 3 routes into IPv6 IS-IS

To redistribute OSPFv3 routes into IPv6 IS-IS, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# redistribute ospf
```

This command configures the device to redistribute all OSPFv3 routes into Level-2 IPv6 IS-IS.

Syntax: [no] **redistribute ospf [level-1 | level-1-2 | level-2 | match external1 | external2 | internal | metric** <*number*> **| metric-type external | internal | route-map** <*name*>]

Most of the parameters are the same as the parameters for the **redistribute static** command. However, the **redistribute ospf** command also has the **match external1** | **external2** | **internal** parameter. This parameter specifies the OSPF route type you want to redistribute into IPv6 IS-IS. By default, the **redistribute ospf** command redistributes only internal routes:

- **external1** – An OSPF type 1 external route.

- **external2** – An OSPF type 2 external route.
- **internal** – An internal route calculated by OSPF.

Redistributing BGP4+ routes into IPv6 IS-IS

To redistribute BGP4+ routes into IPv6 IS-IS, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# redistribute bgp
```

This command configures the router to redistribute all its BGP4 routes into Level-2 IPv6 IS-IS.

Syntax: **[no] redistribute bgp [level-1 | level-1-2 | level-2 | metric** <*number*> **| metric-type external | internal | route-map** <*name*>**]**

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing IPv6 IS-IS routes within IPv6 IS-IS

In addition to redistributing routes from other route sources into IPv6 IS-IS, the device can redistribute Level 1 IPv6 IS-IS routes into Level 2 IPv6 IS-IS routes, and Level 2 IPv6 IS-IS routes into Level 1 IPv6 IS-IS routes. By default, the device redistributes routes from Level 1 into Level 2.

NOTE
The device automatically redistributes Level 1 routes into Level 2 routes, even if you do not enable redistribution.

For example, to redistribute all IPv6 IS-IS routes from Level 2 into Level 1, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# redistribute isis level-2 into level-1
```

The router automatically redistributes Level-1 routes into Level 2.

Syntax: **[no] redistribute isis level-1 into level-2 | level-2 into level-1 [prefix-list** <*name*>**]**

The **level-1 into level-2** | **level-2 into level-1** parameter specifies the direction of the redistribution:

- **level-1 into level-2** – Redistributes Level 1 routes into Level 2. This is the default.
- **level-2 into level-1** – Redistributes Level 2 routes into Level 1.

The optional **prefix-list** <*name*> parameter allows you to specify the IPv6 prefixes that you want redistributed from Level 1 into Level 2 and from Level 2 into Level 1. Specify the name of the IPv6 prefix list that contains the desired prefixes. (For information about prefix lists, including the syntax of the **ipv6 prefix-list** command, refer to Chapter 39, “Configuring an IPv6 Prefix List”.)

For example, to redistribute the IPv6 prefix 2001::/16 from Level 2 into Level 1, enter commands such as the following.

```
NetIron(config)# ipv6 prefix-list routesfor2001 permit 2001::/16
NetIron(config)# router isis
NetIron(config-isis-router)# address-family ipv6 unicast
NetIron(config-isis-router-ipv6u)# redistribute isis level-2 into level-1
prefix-list routesfor2001
```

Disabling and re-enabling IPv6 protocol-support consistency checks

As discussed in "IPv6 IS-IS single-topology mode" on page 1867, an IS-IS single topology must be configured to run the same set of network protocols (IPv4 IS-IS only, IPv6 IS-IS only, or both IPv4 IS-IS and IPv6 IS-IS).

By default, IS-IS performs consistency checks on hello packets. If a hello packet does not have the same configured network protocols, IS-IS rejects the packet. For example, a hello packet from a router running IPv4 and IPv6 IS-IS will be rejected by a router running either IPv4 IS-IS only or IPv6 IS-IS only, and the two routers will not become adjacent.

To allow two routers running different sets of network protocols to form an adjacency, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# no adjacency-check
```

This command disables the IPv6 IS-IS consistency check.

Syntax: [no] adjacency-check

To re-enable the consistency check, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
NetIron(config-isis-router-ipv6u)# adjacency-check
```

Configuring ISIS properties on an interface

The parameter settings for configuring IS-IS properties on a device apply to both IS-IS IPv4 and IS-IS IPv6 except for "Changing the metric added to advertised routes" as described below. For details on how to perform all other IS-IS properties on an Interface, refer to 25, "Configuring IS-IS (IPv4)"

Changing the metric added to advertised routes

When the PowerConnect originates an IS-IS route or calculates a route, the PowerConnect adds a metric (cost) to the route. Each IS-IS interface has a separate metric value. The default is 10.

The PowerConnect applies the interface-level metric to routes originated on the interface and also when calculating routes. The PowerConnect does not apply the metric to link-state information that the PowerConnect receives from one IS and floods to other ISs.

The default interface metric is 10. You can change the metric on an individual interface to a value in one of the following ranges:

- 1 – 63 for the narrow metric style (the default metric style)
- 1 – 16777215 for the wide metric style

NOTE
If the metric value you want to use is higher than 63 but you have not changed the metric style to wide, change the metric style first, then set the metric. The IS-IS neighbors that will receive the advertisements also must be enabled to receive wide metrics.

To change the IS-IS metric on an interface, use the following CLI method.

```
NetIron(config-isis-router)# interface ethernet 2/8
NetIron(config-if-e1000-2/8)# isis metric 44
```

Syntax: [no] isis metric <*num*>

The <*num*> parameter specifies the metric. The range of values you can specify depends on the metric style. You can specify 1 – 63 for the narrow metric style or 1 – 16777215 for the wide metric style. The default in either case is 10.

When IPv6 IS-IS is enabled in a single topology, you must set the metric-style to be wide if you want to use an interface metric greater than 63.

Displaying IPv6 IS-IS information

You can display the following information about IPv6 IS-IS:

- General IPv6 IS-IS information.
- IPv6 IS-IS configuration information.
- IPv6 IS-IS error statistics.
- LSP database entries.
- IS-IS system ID to hostname mappings.
- IPv6 IS-IS interface information.
- IPv6 IS-IS memory usage information.
- IPv6 IS-IS neighbor information.
- IPv6 IS-IS path information.
- IPv6 IS-IS redistribution information.
- IPv6 IS-IS route information.
- IPv6 IS-IS traffic statistics.

Displaying IPv6 IS-IS information

To display general IPv6 IS-IS information, enter the following command at any CLI level.

```
NetIron# show ipv6 isis
 IS-IS  Routing Protocol Operation State: Enabled
 IS-Type: Level-1-2
 System ID: 8888.5555.0008
 Manual area address(es):
   49.8585
 Interfaces with Integrated IS-IS for IPv6 configured:
    Interface 4/1    Interface 4/2    Interface 4/11    Interface 4/12
    Interface 4/13   Interface 4/14   Interface 4/15    Interface 4/16
    Interface 4/17   Interface 4/35   Interface 4/36    Interface 4/37
    Interface 4/38   Interface v43    Interface v44     Interface lb1
 Following Routes are Redistributed into IS-IS for IPv6:
   CONNECTED
 Number of Routes redistributed into IS-IS: 1
 Domain password: None
 Area password: None
 IS-IS for IPV6 Route Administrative Distance: 115
 Hold Time Between Two SPF Calculations: 5
 Global Hello Padding: Enabled
```

Syntax: show ipv6 isis

This display shows the following information.

TABLE 336 IPv6 IS-IS information fields

This field...	Displays...
IS-IS Routing Protocol Operation State	The operating state of IPv6 IS-IS. Possible states include the following: • Enabled – IPv6 IS-IS is enabled. • Disabled – IPv6 IS-IS is disabled.
IS Type	The intermediate system type. Possible types include the following: • Level 1 only – The device routes traffic only within the area in which it resides. • Level 2 only – The device routes traffic between areas of a routing domain. • Level 1-2 – The device routes traffic within the area in which it resides and between areas of a routing domain.
System ID	The unique IS-IS router ID. Typically, the device's base MAC address is used as the system ID.
Manual area address(es)	Area address(es) of the device.
Interfaces with Integrated IS-IS for IPv6 configured	Interfaces on which IPv6 IS-IS is configured.
Following Routes are Redistributed into IS-IS for IPv6	Routes that are redistributed into IPv6 IS-IS. Possible routes include the following: • BGP – BGP4+ routes are redistributed into IPv6 IS-IS. • RIP – RIPng routes are redistributed into IPv6 IS-IS. • OSPF – OSPFv3 routes are redistributed into IPv6 IS-IS. • STATIC – Static IPv6 routes are redistributed into IPv6 IS-IS. • CONNECTED – IPv6 routes learned from directly connected networks are redistributed into IPv6 IS-IS.

TABLE 336 IPv6 IS-IS information fields (Continued)

This field...	Displays...
Number of Routes redistributed into IS-IS	The number of routes distributed into IS-IS
Domain password	The domain password, if one is configured.
Area password	The domain password, if one is configured.
IS-IS IPv6 Route Administrative Distance	The current setting of the IPv6 IS-IS administrative distance.
Hold Time Between Two SPF Calculations	The setting of the SPF timer, which causes the router to recalculate the SPF tree of its IPv6 IS-IS links following a change in topology or the link state database.
Global Hello Padding	The setting of the global hello padding feature, which can be one of the following: • Disabled – Global padding for hello packets is disabled. • Enabled – Global padding for hello packets is enabled.

Displaying the IPv6 IS-IS configuration in the running configuration

You can display the IPv6 IS-IS commands that are in effect on the device.

NOTE
The running configuration does not list the default values. Only commands that change a setting or add configuration information are displayed.

To display the IPv6 IS-IS configuration, enter the following command at any CLI level.

```
NetIron# show ipv6 isis config
Current ISIS configuration:
router isis
 net  49.6561.2222.2222.2222.00

 address-family ipv4 unicast
 distance  135
 redistribute static
 exit-address-family

 address-family ipv6 unicast
 redistribute static
 exit-address-family

end
```

Syntax: show ipv6 isis config

The running configuration shown in this example contains the following commands:

- Global IPv6 IS-IS commands that enable IS-IS.
- Address family commands that configure IPv4 IS-IS unicast routes.
- Address family commands that configure IPv6 IS-IS unicast routes.

Displaying IPv6 IS-IS error statistics

To display IPv6 IS-IS error statistics, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis counts
Area Mismatch: 0
Max Area Mismatch: 0
System ID Length Mismatch: 0
Authentication Fail: 0
Corrupted LSP: 0
LSP Sequence Number Skipped: 0
LSP Max Sequence Number Exceeded: 0
Level-1 Database Overload: 0
Level-2 Database Overload: 0
Our LSP Purged: 0
```

Syntax: show ipv6 isis counts

This display shows the following information.

TABLE 337 IPv6 IS-IS error statistics

This field...	Displays...
Area Mismatch	The number of times the router interface was unable to create a Level-1 adjacency with a neighbor because the router interface and the neighbor did not have any areas in common.
Max Area Mismatch	The number of times the device received a PDU with a value for maximum number of area addresses that did not match the device's value for maximum number of area addresses.
System ID Length Mismatch	The number of times the device received a PDU with an ID field that was a different length than the ID field length configured on the router.
Authentication Fail	The number of times authentication failed because the device was configured to authenticate IPv6 IS-IS packets in the packet's domain or area, but the packet did not contain the correct password.
Corrupted LSP	The number of times the device detected a corrupted LSP in the device's memory.
LSP Sequence Number Skipped	The number of times the device received an LSP with a sequence number that was more than 1 higher than the sequence number of the previous LSP received from the same neighbor.
LSP Max Sequence Number Exceeded	The number of times the device attempted to set an LSP sequence number to a value higher than the highest number in the CSNP sent by the Designated IS.
Level-1 Database Overload	The number of times the Level-1 state on the router changed from Waiting to On or from On to Waiting: • Waiting to On – This change can occur when the device recovers from a previous Level-1 LSP database overload and is again ready to receive new LSPs. • On to Waiting – This change can occur when the device's Level-1 LSP database is full and the device receives an additional LSP, for which there is no room.

TABLE 337 IPv6 IS-IS error statistics (Continued)

This field...	Displays...
Level-2 Database Overload	The number of times the Level-2 state on the device changed from Waiting to On or from On to Waiting: • The change from Waiting to On can occur when the device recovers from a previous Level-2 LSP database overload and is again ready to receive new LSPs. • The change from On to Waiting can occur when the device's Level-2 LSP database is full and the device receives an additional LSP, for which there is no room.
Our LSP Purged	The number of times the device received an LSP that was originated by the device itself and had age zero (aged out).

Displaying LSP database entries

You can display summary or detailed information about the entries in the LSP database.

NOTE
The router maintains separate LSP databases for Level 1 LSPs and Level 2 LSPs.

To display summary information about the entries in the LSP database, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis database
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
Router1.00-00         0x00000003   0x9a6b        574               0/0/0
Router2.00-00*        0x00000002   0x609d        540               0/0/0
Router2.01-00*        0x00000001   0x0fcf        539               0/0/0

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
Router1.00-00         0x00000003   0xe2da        574               0/0/0
Router2.00-00*        0x00000002   0x0585        540               0/0/0
Router2.01-00*        0x00000001   0x0fcf        539               0/0/0
```

The command in this example displays information for the LSPS in the router's Level-1 and Level-2 LSP databases. Notice that the display groups the Level-1 and Level-2 LSPs separately.

Syntax: show ipv6 isis database [<HHHH.HHHH.HHHH.HH-HH> | detail | l1 | l2 | level1 | level2]

The <HHHH.HHHH.HHHH.HH-HH> parameter restricts the display to the entry for the specified LSPID. (The LSPID consists of the source ID (HHHH.HHHH.HHHH), the pseudonode (HH-), and LSPID (-HH). To determine the router's source ID, use the **show ipv6 isis** command. For more information, refer to "Displaying IPv6 IS-IS information" on page 1881. To determine the pseudonode and LSPID, use the **show ipv6 isis database** command.

NOTE
Name mapping is enabled by default. When name mapping is enabled, the output of the **show ipv6 isis database** command uses the hostname instead of the system ID. To disable mapping so that these displays use the system ID instead, enter the **no hostname** command at the IS-IS router configuration level.

The **detail** parameter displays detailed information about the LSPs. The detailed information display is discussed later in this section.

The **l1** and **level1** parameters restrict the display to the Level-1 LSP entries. You can use these parameters interchangeably.

The **l2** and **level2** parameters restrict the display to the Level-2 LSP entries. You can use these parameters interchangeably.

This display shows the following information.

TABLE 338 IPv6 IS-IS summary LSP database information

This field...	Displays...
LSPID	The LSP ID, which consists of the source ID (HHHH.HHHH.HHHH), the pseudonode (HH-), and LSPID (-HH). **Note**: If the address has an asterisk (*) at the end, this indicates that the LSP is locally originated.
LSP Seq Num	The sequence number of the LSP.
LSP Checksum	The checksum calculated by the device that sent the LSP and used by the device to verify that the LSP was not corrupted during transmission over the network.
LSP Holdtime	The maximum number of seconds during which the LSP will remain valid. **Note**: The IS that originates the LSP starts the timer for the LSP. As a result, LSPs do not all have the same amount of time remaining when they enter the device's LSP database.
ATT	A 4-bit value extracted from bits 4 – 7 in the Attach field of the LSP.
P	The value in the Partition option field of the LSP. The field can have one of the following values: • 0 – The IS that sent the LSP does not support partition repair. • 1 – The IS that sent the LSP supports partition repair.
OL	The value in the LSP database overload field of the LSP. The field can have one of the following values: • 0 – The overload bit is off. • 1 – The overload bit is on, indicating that the IS that sent the LSP is overloaded and should not be used as a Level-2 router.

To display detailed information for all the LSPs in the device's LSP databases, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis database detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
Router1.00-00         0x00000003   0x9a6b        566               0/0/0
   Area Address:  49.6561
   Metric:  10    IS Router2.01
   Metric:  12    IS Router1.02
   Metric:  10    IS Router1.03
   NLPID:  8e  cc
   IP address:  10.0.0.1
   Metric:  10    IP-Internal 110.10.0.0       255.255.0.0
   Metric:  10    IP-Internal 110.20.0.0       255.255.0.0
   Metric:  10    IP-Internal 110.30.0.0       255.255.0.0
    ...
     Hostname:  Router1
   IPv6 address:  3001::1
   Metric:  10    IPv6 Reachability 1111:5000::/32  UP bit: 0
   Metric:  10    IPv6 Reachability 1111:4000::/32  UP bit: 0
   Metric:  10    IPv6 Reachability 1111:3000::/32  UP bit: 0
...
```

NOTE
Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

Syntax: show ipv6 isis database detail [l1 | l2 | level1 | level2]

The **l1** and **level1** parameters restrict the display to the Level-1 LSP entries. You can use these parameters interchangeably.

The **l2** and **level2** parameters restrict the display to the Level-2 LSP entries. You can use these parameters interchangeably.

For example, to display details about Level-1 LSPs only, enter a command such as the following at any CLI level.

```
NetIron# show ipv6 isis database detail l1
```

This display shows the following information.

TABLE 339 IPv6 IS-IS detailed LSP database information

This field...	Displays...
LSPID	Refer to the description in Table 338 on page 1885.
LSP Seq Num	Refer to the description in Table 338 on page 1885.
LSP Checksum	Refer to the description in Table 338 on page 1885.
LSP Holdtime	Refer to the description in Table 338 on page 1885.
ATT/P/OL	Refer to the description in Table 338 on page 1885.
Area Address	The address of the area.

TABLE 339 IPv6 IS-IS detailed LSP database information (Continued)

This field...	Displays...
TLVs	The remaining output displays the type, length, and value (TLV) parameters included in the LSPs. These parameters advertise reachability to IPv6 devices or networks. For example: • A router identified as an IS and its hostname (Router2.01) can be reached using the default metric of 10. • An end system within the current area identified as an IP-Internal and with the IP address of 110.10.0.0 and sub-net mask of 255.255.0.0 can be reached using the default metric of 10. • An IPv6 prefix of 1111:5000::/32 is up and can be reached using the default metric of 10.
NLPID	The Network Layer Protocol Identifier (NLPID), which specifies the protocol the IS that sent the LSP is using. Usually, this value is “cc” but can also be “iso”.
IP address	The IP address of the interface that sent the LSP. The device can use this address as the next hop in routes to the addresses listed in the rows below.
Hostname	The hostname of the router that contains the LSP database that is displayed.
IPv6 address	The IPv6 address of the interface that sent the LSP. The device can use this address as the next hop in routes to the addresses listed in the rows below.

Displaying the system ID to name mappings

IS-IS maps the IS-IS system IDs to the hostnames of the devices with those IS. To display these mappings, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis hostname
Total number of entries in IS-IS Hostname Table: 2
   System ID       Hostname            * = local IS
 * 2222.2222.2222  Router2
   1111.1111.1111  Router1
```

Syntax: show ipv6 isis hostname

This example contains two mappings for this device. The device’s IS-IS system ID is “2222.2222.2222“and its hostname is “Router2”. The display contains an entry for another router. The display contains one entry for each IS that supports name mapping.

NOTE
Name mapping is enabled by default. When name mapping is enabled, the output of the **show ipv6 isis database** and **show ipv6 isis neighbor** commands uses the hostname instead of the system ID. To disable mapping so that these displays use the system ID instead, enter the **no hostname** command at the IS-IS router configuration level.

Displaying IPv6 IS-IS interface information

To display information about the interfaces on which IPv6 IS-IS is enabled, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis interface
Total number of IS-IS Interfaces: 4

Interface : 2/1     Local Circuit Number: 00000001
   Circuit Type : BCAST Circuit Mode : LEVEL-1-2
   Circuit State: UP Passive State: FALSE
   MTU : 1497
   Level-1 Metric: 10, Level-1 Priority: 64
   Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
   Level-1 Designated IS: Router2.01-22         Level-1 DIS Changes: 8
   Level-2 Metric: 10, Priority: 64
   Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
   Level-2 Designated IS: Router2.01-00 Level-2 DIS Changes: 8
   Next IS-IS LAN Level-1 Hello in 1 seconds
   Next IS-IS LAN Level-2 Hello in 1 seconds
   Number of active Level-1 adjacencies: 1
   Number of active Level-2 adjacencies: 1
  Circuit State Changes: 0 Circuit Adjacencies State Changes: 2
   Rejected Adjacencies: 0
   Circuit Authentication Fails: 0 Bad LSP 0
   Control Messages Sent: 1696 Control Messages Received: 159
   IP Enabled: TRUE
   IP Address and Subnet Mask:
      10.0.0.2            255.0.0.0
     192.147.201.150     255.255.255.0
  IPv6 Enabled: TRUE
   IPv6 Address :
     3001::2
. . .
```

NOTE
The latter part of this display is truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

Syntax: show ipv6 isis interface

This display shows the following information.

TABLE 340 IPv6 IS-IS interface information

This field...	Displays...
Total number of IS-IS interfaces	The number of interfaces on which IPv6 IS-IS is enabled.
Interface	The port or virtual interface number to which the information listed below applies.
Local Circuit Number	The ID that the instance of IPv6 IS-IS running on the interface applied to the circuit between this interface and the interface at the other end of the link.
Circuit Type	The type of IS-IS circuit running on the interface. The circuit type can be one of the following: • BCAST– broadcast • PTP – point-to-point

TABLE 340 IPv6 IS-IS interface information (Continued)

This field...	Displays...
Circuit Mode	The IS-IS type in use on the circuit. The mode can be one of the following: • LEVEL-1 • LEVEL-2 • LEVEL-1-2
Circuit State	The state of the circuit, which can be one of the following: • DOWN • UP
Passive State	The state of the passive option, which determines whether the interface is allowed to form an IS-IS adjacency with the IS at the other end of the circuit. The state can be one of the following: • FALSE – The passive option is disabled. The interface can form an adjacency with the IS at the other end of the link. • TRUE – The passive option is enabled. The interface cannot form an adjacency, but can still advertise itself into the area.
MTU	The maximum length supported for IS-IS PDUs sent on this interface.
Level-1 Metric	The default-metric value that the device inserts in IS-IS Level-1 PDUs originated on this interface.
Level-1 Priority	The priority of this IS to be elected as the Designated IS for Level-1 in this broadcast network.
Level-1 Hello Interval	The number of seconds the software waits between sending Level-1 hello PDUs to the IS at the other end of the circuit.
Level-1 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time for Level-1 Hello messages received on the circuit.
Level-1 Designated IS	The NET of the Level-1 Designated IS.
Level-1 DIS Changes	The number of times the NET of the Level-1 Designated IS has changed.
Level-2 Metric	The default-metric value that the router inserts in IS-IS Level-2 PDUs originated on this interface.
Level-2 Priority	The priority of this IS to be elected as the Designated IS for Level-2 in this broadcast network.
Level-2 Hello Interval	The number of seconds the software waits between sending Level-2 Hello messages to the IS at the other end of the circuit.
Level-2 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time for Level-2 LSPs received on the circuit.
Level-2 Designated IS	The NET of the Level-2 Designated IS.
Level-2 DIS Changes	The number of times the NET of the Level-2 Designated IS has changed.
Next IS-IS LAN Level-1 Hello	Number of seconds before next Level-1 Hello message will be transmitted by the device.
Next IS-IS LAN Level-2 Hello	Number of seconds before next Level-2 Hello message will be transmitted by the device.
Number of active Level-1 adjacencies	The number of ISs with which this interface has an active Level-1 adjacency.
Number of active Level-2 adjacencies	The number of ISs with which this interface has an active Level-2 adjacency.
Circuit State Changes	The number of times the state of the circuit has changed.

TABLE 340 IPv6 IS-IS interface information (Continued)

This field...	Displays...
Circuit State Adjacencies Changes	The number of times an adjacency has started or ended on this circuit.
Rejected Adjacencies	The number of adjacency attempts by other ISs rejected by the router.
Circuit Authentication Fails	The number of times the device rejected a circuit because the authentication did not match the authentication configured on the device.
Bad LSP	The number of times the interface received a bad LSP from an IS at the other end of the circuit. The following conditions can cause an LSP to be bad: • Invalid checksum • Invalid length • Invalid lifetime value
Control Messages Sent	The number of IS-IS control PDUs sent on this interface.
Control Messages Received	The number of IS-IS control PDUs received on this interface.
IP Enabled	The state of IP on the interface, which can be one of the following: • TRUE – IP is enabled. • FALSE – IP is disabled.
IP Address and Subnet Mask	The IP address(es) and sub-net masks configured on this interface.
IPv6 Enabled	The state of IPv6 on the interface, which can be one of the following: • TRUE – IPv6 is enabled. • FALSE – IPv6 is disabled.
IPv6 Address	The IPv6 address(es) configured on this interface.

Displaying IPv6 IS-IS memory usage

To display information about IPv6 IS-IS memory usage, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis memory
Total Static Memory Allocated : 1333 bytes
Total Dynamic Memory Allocated : 157952 bytes
Memory Type               Size       Allocated  Max-alloc  Alloc-Fails
MTYPE_ISIS_IP6_SUMMARY_PR 0          0          0          0
MTYPE_ISIS_OTHER          20         0          1          0
MTYPE_ISIS_IP6_ROUTE_NODE 21         22         1024       0
MTYPE_ISIS_IP6_ROUTE_INFO 12         17         1024       0
MTYPE_ISIS_IP6_NEXTHOP    24         2          256        0
MTYPE_ISIS_IP6_REDIS_ROUT 12         5          256        0
```

Syntax: show ipv6 isis memory

This display shows the following information.

TABLE 341 IPv6 IS-IS memory usage information

This field...	Displays...
Total Static Memory Allocated	A summary of the amount of static memory allocated, in bytes, to IPv6 IS-IS.
Total Dynamic Memory Allocated	A summary of the amount of dynamic memory allocated, in bytes, to IPv6 IS-IS.

TABLE 341 IPv6 IS-IS memory usage information (Continued)

This field...	Displays...
Memory Type	The type of memory used by IPv6 IS-IS. (This information is for use by Dell technical support in case of a problem.)
Size	The size of a memory type.
Allocated	The amount of memory currently allocated to a memory type.
Max-alloc	The maximum amount of memory that was allocated to a memory type.
Alloc-Fails	The number of times an attempt to allocate memory to a memory type failed.

Displaying IPv6 IS-IS neighbor information

You can display a summary or detailed information for all neighbors with which the device has formed an IS-IS adjacency.

To display a summary of all IPv6 IS-IS neighbors of a router, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis neighbor
Total number of IS-IS Neighbors: 2
System Id      Interface   SNPA           State Holdtime Type Pri StateChgeTime
Router1        Ether 3/2   00e0.5200.0020 UP    30       ISL2 64  0  :0 :14:1
Router1        Ether 3/2   00e0.5200.0020 UP    30       ISL1 64  0  :0 :14:1
```

Syntax: show ipv6 isis neighbor [detail]

This display shows the following information.

TABLE 342 Summary of IPv6 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	The number of ISs with which the device has formed an IS-IS adjacency.
System ID	The system ID of the neighbor. **Note:** Name mapping is enabled by default. When name mapping is enabled, the output of the **show ipv6 isis neighbor** command uses the hostname instead of the system ID. To disable mapping so that these displays use the system ID instead, enter the **no hostname** command at the IS-IS router configuration level. For more information about performing this task, refer to the " 25, "Configuring IS-IS (IPv4)"" .
Interface	The router port or virtual interface attached to the neighbor.
SNPA	The Subnetwork Point of Attachment (SNPA), which is the MAC address of the device physical or virtual interface attached to the neighbor.
State	The state of the adjacency with the neighbor. The state can be one of the following: • DOWN – The adjacency is down. • INIT – The adjacency is being established and is not up yet. • UP – The adjacency is up.
Holdtime	The time between transmissions of IS-IS hello messages.

TABLE 342 Summary of IPv6 IS-IS neighbor information (Continued)

This field...	Displays...
Type	The IS-IS type of the adjacency. The type can be one of the following: • ISL1 – Level-1 IS • ISL2 – Level-2 IS • PTP – Point-to-Point IS • ES – ES **Note**: The device forms a separate adjacency for each IS-IS type. Thus, if the router has both types of IS-IS adjacencies with the neighbor, the display contains a separate row of information for each adjacency.
Pri	The priority of this IS to be elected as the Designated IS in this broadcast network.
StateChgeTime	The amount of time that has passed since the adjacency last changed state.

To display detailed information about all IPv6 IS-IS neighbors of a router, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis neighbor detail
Total number of IS-IS Neighbors: 2
System ID      Interface   SNPA           State Holdtime Type Pri StateChgeTime
Router1        Ether 3/2   00e0.5200.0020 UP    30       ISL2 64  0  :0 :14:5
Area Address(es): 49.6561
IP Address(es): 10.0.0.1
IPv6 Address: fe80::2e0:52ff:fe00:20
Circuit ID: 2222.2222.2222.01
System ID      Interface   SNPA           State Holdtime Type Pri StateChgeTime
Router1        Ether 3/2   00e0.5200.0020 UP    30       ISL1 64  0  :0 :14:5
Area Address(es): 49.6561
IP Address(es): 10.0.0.1
IPv6 Address: fe80::2e0:52ff:fe00:20
Circuit ID: 2222.2222.2222.01
```

This display shows the following information.

TABLE 343 Detailed IPv6 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	For information about this field, refer to Table 342 on page 1891.
System ID	For information about this field, refer to Table 342 on page 1891.
Interface	For information about this field, refer to Table 342 on page 1891.
SNPA	For information about this field, refer to Table 342 on page 1891.
State	For information about this field, refer to Table 342 on page 1891.
Holdtime	For information about this field, refer to Table 342 on page 1891.
Type	For information about this field, refer to Table 342 on page 1891.
Pri	For information about this field, refer to Table 342 on page 1891.
StateChgeTime	For information about this field, refer to Table 342 on page 1891.
Area Address(es)	The address(es) of areas to which the neighbor interface belongs.
IP Address(es)	The IP address(es) assigned to the neighbor interface.

TABLE 343 Detailed IPv6 IS-IS neighbor information (Continued)

This field...	Displays...
IPv6 Address	The IPv6 address(es) assigned to the neighbor interface.
Circuit ID	The ID of the IS-IS circuit running on the neighbor interface.

Displaying IPv6 IS-IS redistribution information

To display information about the IPv6 routes redistributed into IPv6 IS-IS, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis redistributed-routes
Prefix                                    Protocol  Level        Metric
5555:1002::/32                            Static    Level-2       1
5555:2002::/32                            Static    Level-2       1
5555:3002::/32                            Static    Level-2       1
5555:4002::/32                            Static    Level-2       1
5555:5002::/32                            Static    Level-2       1
```

Syntax: show ipv6 isis redistributed-routes

This display shows the following information.

TABLE 344 IPv6 IS-IS redistribution information

This field...	Displays...
Prefix	The IPv6 routes redistributed into IPv6 IS-IS.
Protocol	The protocol from which the route is redistributed into IPv6 IS-IS. Possible protocols include the following: • BGP – BGP4+. • RIP – RIPng. • OSPF – OSPFv3. • Static – IPv6 static route table. • Connected – A directly connected network.
Level	The IS-IS level into which a route is redistributed. Possible levels include the following: • Level-1 • Level-2 • Level-1-2
Metric	The value of the default redistribution metric, which is the IS-IS cost of redistributing the route into IPv6 IS-IS.

Displaying the IPv6 IS-IS route information

To display the routes in the router's IPv6 IS-IS route table, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis routes
ISIS IPv6 Routing Table
Total Routes: 17  Level1: 17 Level2: 0 Equal-cost multi-path: 0
Type IPv6 Prefix                Next Hop Router           Interface  Cost
L1   1111:1000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   1111:2000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   1111:3000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   1111:4000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   1111:5000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   2222:1000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   30
L1   2222:2000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   30
L1   2222:3000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   30
L1   2222:4000::/32             fe80::2e0:52ff:fe00:20    ethe 3/2   30
```

Syntax: show ipv6 isis routes

This display shows the following information.

TABLE 345 IPv6 IS-IS route information

This field...	Displays...
Total Routes	The total number of routes in the router's IPv6 IS-IS route table. The total includes Level-1 and Level-2 routes.
Level1	The total number of Level-1 routes in the IPv6 IS-IS route table.
Level2	The total number of Level-1 routes in the IPv6 IS-IS route table.
Equal-cost multi-path	The total number of equal-cost routes to the same destination in the IPv6 IS-IS route table. If load sharing is enabled, the router equally distributes traffic among the routes.
Type	The route type, which can be one of the following: • L1 – Level-1 route • L2 – Level-2 route
IPv6 Prefix	The IPv6 prefix of the route.
Next Hop Router	The IPv6 address of the next-hop interface to the destination.
Interface	The router interface (physical or virtual interface) attached to the next hop.
Cost	The IPv6 IS-IS default metric for the route, which is the cost of using this route to reach the next-hop router to this destination.

Displaying IPv6 IS-IS traffic statistics

The router maintains statistics for common IS-IS PDU types. To display the IPv6 traffic statistics, enter the following command at any level of the CLI.

```
NetIron# show ipv6 isis traffic
                        Message Received     Message Sent
Level-1 Hellos            98                   1171
Level-2 Hellos            96                   1170
PTP Hellos                0                    0
Level-1 LSP               3                    6
Level-2 LSP               3                    6
Level-1 CSNP              1                    110
Level-2 CSNP              1                    110
Level-1 PSNP              0                    0
Level-2 PSNP              0                    0
```

Syntax: show ipv6 isis traffic

This display shows the following information.

TABLE 346 IPv6 IS-IS traffic statistics

This field...	Displays...
Level-1 Hellos	The number of Level-1 hello PDUs sent and received by the router.
Level-2 Hellos	The number of Level-2 hello PDUs sent and received by the router.
PTP Hellos	The number of point-to-point hello PDUs sent and received by the router.
Level-1 LSP	The number of Level-1 link-state PDUs sent and received by the router.
Level-2 LSP	The number of Level-2 link-state PDUs sent and received by the router.
Level-1 CSNP	The number of Level-1 Complete Sequence Number PDUs (CSNPs) sent and received by the router.
Level-2 CSNP	The number of Level-2 CSNPs sent and received by the router.
Level-1 PSNP	The number of Level-1 Partial Sequence Number PDUs (PSNPs) sent and received by the router.
Level-2 PSNP	The number of Level-2 PSNPs sent and received by the router.

Chapter

Configuring BGP4+ 45

PowerConnect B-MLXe supports the folllowing BGP+ features:

- BGP+
- Configuring BGP4+ Neighbors Using Global or Site-Local IPv6 Addresses
- Importing Routes into BGP4+
- Advertising the Default BGP4+ Route
- Clearing BGP4+ Information
- Displaying BGP4+ Information

The implementation of IPv6 supports multi protocol BGP (MBGP) extensions, which allow IPv6 BGP (known as ***BGP4+***) to distribute routing information for protocols such as IPv4 BGP. The supported protocols are identified by address families. (For information about address families, refer to "Address family configuration level" on page 1897 and Appendix B, "Global and Address Family Configuration Levels".) The extensions allow a set of BGP4+ peers to exchange routing information for multiple address families and sub-address families.

IPv6 MBGP functions similarly to IPv4 MBGP except for the following enhancements:

- An IPv6 unicast address family and network layer reachability information (NLRI).
- Next hop attributes that use IPv6 addresses.

NOTE
The implementation of BGP4+ supports the advertising of routes among different address families. However, it supports BGP4+ unicast routes only; it does not currently support BGP4+ multicast routes.

Address family configuration level

The implementation of BGP4+ includes a new configuration level: address family. For IPv6, the PowerConnect devices currently supports the BGP4+ unicast address family configuration level only. (For IPv4, the PowerConnect devices supports the BGP4 unicast and BGP4 multicast address family configuration levels.) The enters the BGP4+ unicast address family configuration level when you enter the following command while at the global BGP configuration level:

```
NetIron(config-bgp)# address-family ipv6 unicast
NetIron(config-bgp-ipv6u)#
```

The `(config-bgp-ipv6u)#` prompt indicates that you are at the BGP4+ unicast address family configuration level.

While at the BGP4+ unicast address family configuration level, you can access several commands that allow you to configure BGP4+ unicast routes. The commands that you enter at this level apply only to IPv6 unicast address family only. You can generate a configuration for BGP4+ unicast routes that is separate and distinct from configurations for IPv4 unicast routes and IPv4 BGP multicast routes.

NOTE
The commands that you can access while at the IPv6 unicast address family configuration level are also available at the IPv4 unicast and multicast address family configuration levels. Where relevant, this section discusses and provides IPv6-unicast-specific examples. You must first configure IPv6 unicast-routing in order for any IPv6 routing protocol to be active.

NOTE
Each address family configuration level allows you to access commands that apply to that particular address family only. To enable a feature in a particular address family, you must specify any associated commands for that feature in that particular address family. You cannot expect the feature, which you may have configured in the BGP4 unicast address family, to work in the BGP4+ unicast address family unless it is explicitly configured in the BGP4+ unicast address family.

To exit from the IPv6 unicast address family configuration level, enter the following command:

```
NetIron(config-bgp-ipv6u)# exit-address-family
NetIron(config-bgp)#
```

Entering this command returns you to the global BGP configuration level.

For complete information about the new CLI levels, refer to Appendix B, "Global and Address Family Configuration Levels".

Configuring BGP4+

Before enabling BGP4+ on a , you must enable the forwarding of IPv6 traffic on the using the **ipv6 unicast-routing** command and enable IPv6 on at least one interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface. For more information on performing these configuration tasks, refer to 38, "Configuring Basic IPv6 Connectivity".

To configure BGP4+, you must do the following:

- Enable BGP4+.
- Configure BGP4+ neighbors using one of the following methods:
  - Add one neighbor at a time (neighbor uses global or site-local IPv6 address).
  - Add one neighbor at a time (neighbor uses a link-local IPv6 address).
  - Create a peer group and add neighbors individually.

The following configuration tasks are optional:

- Advertise the default route.
- Import specified routes into BGP4+.
- Redistribute prefixes into BGP4+.
- Aggregate routes advertised to BGP4 neighbors.
- Use route maps.

Enabling BGP4+

To enable BGP4+, enter commands such as the following:

```
NetIron(config)# router bgp
BGP: Please configure 'local-as' parameter in order to run BGP4.
NetIron(config-bgp)# local-as 1000
```

These commands enables the BGP4+ router and configure the autonomous system (1000) in which your resides.

Syntax: [no] router bgp

To disable BGP, enter the **no** form of this command.

Syntax: local-as <*number*>

Specify the AS number in which the you are configuring resides.

After enabling BGP4+, you can add neighbors to a BGP4+ router by entering a commands such as the following:

```
NetIron(config-bgp)# address-family ipv6 unicast
NetIron(config-bgp-ipv6u)# neighbor 2001:4383:e0ff:783a::4 remote-as 1001
NetIron(config-bgp-ipv6u)# neighbor 2001:4383:e0ff:783a::5 remote-as 1001
```

These commands add two neighbors with global IPv6 addresses 2001:4383:e0ff:783a::4 and 2001:4383:e0ff:783a::5 in AS 1001.

NOTE
The example above adds IPv6 neighbors at the BGP4+ unicast address family configuration level. These neighbors, by default, are enabled to exchange BGP4+ unicast prefixes. However, if you add IPv6 neighbors while at the global BGP configuration or IPv4 BGP unicast address family configuration level, the neighbors will not exchange BGP4+ unicast prefixes until you explicitly enable them to do so by entering the **neighbor** <*ipv6-address*> | <*peer-group-name*> **activate** command at the BGP4+ unicast address family configuration level.

This section provides minimal information about adding BGP4+ neighbors, because its focus is to provide information about configuring BGP4+.

Configuring BGP4+ neighbors using global or site-local IPv6 addresses

To configure BGP4+ neighbors using global or site-local IPv6 addresses, you must add the IPv6 address of a neighbor in a remote AS to the BGP4+ neighbor table of the local . You must repeat this procedure for each neighbor that you want to add to a local .

For example, to add the IPv6 address 2011:f3e9:93e8:cc00::1 of a neighbor in remote AS 4500 to the BGP4+ neighbor table of a , enter the following commands:

```
NetIron(config-bgp)# address-family ipv6 unicast
NetIron(config-bgp-ipv6u)# neighbor 2011:f3e9:93e8:cc00::1 remote-as 4500
```

Syntax: neighbor <*ipv6-address*> **remote-as** <*as-number*>

NOTE
The example above adds an IPv6 neighbor at the BGP4+ unicast address family configuration level. This neighbor, by default, is enabled to exchange BGP4+ unicast prefixes. However, if you add an IPv6 neighbor while at the global BGP configuration or IPv4 BGP unicast address family configuration level, the neighbor will not exchange BGP4+ unicast prefixes until you explicitly enable it to do so by entering the **neighbor** <*ipv6-address*> | <*peer-group-name*> activate command at the BGP4+ unicast address family configuration level.

The **ipv6-address** parameter specifies the IPv6 address of the neighbor. You must specify the **ipv6-address** parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **as-number** parameter indicates the number of the AS in which the neighbor resides.

To delete the neighbor from the BGP4+ neighbor table, enter the **no** form of this command.

Adding BGP4+ neighbors using link-local addresses

To configure BGP4+ neighbors that use link-local addresses, you must do the following:

- Add the IPv6 address of a neighbor in a remote AS to the BGP4+ neighbor table of the local .
- Identify the neighbor interface over which the neighbor and local will exchange prefixes.
- Configure a route map to set up a global next hop for packets destined for the neighbor.

Adding BGP4+ neighbor

To add the IPv6 link-local address fe80:4398:ab30:45de::1 of a neighbor in remote AS 1000 to the BGP4+ neighbor table of a , enter the following commands:

```
NetIron(config-bgp)# address-family ipv6 unicast
NetIron(config-bgp-ipv6u)# neighbor fe80:4398:ab30:45de::1 remote-as 1000
```

Syntax: neighbor <*ipv6-address*> **remote-as** <*as-number*>

NOTE
The example above adds an IPv6 neighbor at the BGP4+ unicast address family configuration level. This neighbor, by default, is enabled to exchange BGP4+ unicast prefixes. However, if you add an IPv6 neighbor while at the global BGP configuration or IPv4 BGP unicast address family configuration level, the neighbor will not exchange BGP4+ unicast prefixes until you explicitly enable it to do so by entering the **neighbor** <*ipv6-address*> | <*peer-group-name*> **activate** command at the BGP4+ unicast address family configuration level.

The <*ipv6-address*> parameter specifies the IPv6 link-local address of the neighbor. A link-local address has a fixed prefix of FE80::/10. You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*as-number*> parameter indicates the number of the AS in which the neighbor resides.

To delete the neighbor from the BGP4+ neighbor table, enter the **no** form of this command.

Identifying a neighbor interface

To specify Ethernet interface 3/1 as the neighbor interface over which the neighbor and local will exchange prefixes, enter the following command:

```
NetIron(config-bgp)# neighbor fe80:4398:ab30:45de::1 update-source ethernet 3/1
```

Syntax: neighbor <*ipv6-address*> **update-source** <*ipv4-address*> **| ethernet** <*port*> **| loopback** <*number*> **| ve** <*number*>

The <*ipv6-address*> parameter specifies the IPv6 link-local address of the neighbor. A link-local address has a fixed prefix of FE80::/10. You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*ipv4-address*> parameter specifies the IPv4 address of the update source.

The **ethernet | loopback | ve** parameter specifies the neighbor interface over which the neighbor and local will exchange prefixes. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback or VE interface, also specify the loopback or VE number.

Configuring a route map

To configure a route map that filters routes advertised to a neighbor or sets up a global next hop for packets destined for the neighbor with the IPv6 link-local address fe80:4393:ab30:45de::1, enter commands such as the following (start at the BGP4+ unicast address family configuration level):

```
NetIron(config-bgp-ipv6u)# neighbor fe80:4398:ab30:45de::1 route-map out next-hop
NetIron(config-bgp-ipv6u)# exit
NetIron(config)# route-map next-hop permit 10
NetIron(config-route-map)# match ipv6 address prefix-list next-hop-ipv6
NetIron(config-route-map)# set ipv6 next-hop 2011:e0ff:3764::34
```

This route map applies to the BGP4+ unicast address family under which the **neighbor route-map** command is entered. This route map applies to the outgoing routes on the neighbor with the IPv6 link-local address fe80:4393:ab30:45de::1. If an outgoing route on the neighbor matches the route map, the route is distributed through the next hop router with the global IPv6 address 2011:e0ff:3764::34.

Syntax: neighbor <*ipv6-address*> **route-map [in | out]** <*name*>

The <*ipv6-address*> parameter specifies the IPv6 link-local address of the neighbor. A link-local address has a fixed prefix of FE80::/10. You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **in** keyword applies the route map to incoming routes. The **out** keyword applies the route map to outgoing routes.

The <*name*> parameter specifies a route map name.

Syntax: route-map <*name*> **deny | permit** <*sequence-number*>

The **name** parameter specifies a route map name.

The **deny** keyword denies the distribution of routes that match the route map. The **permit** keyword permits the distribution of routes that match the route map.

The <*sequence-number*> parameter specifies a sequence number for the route map statement.

Syntax: match ipv6 address prefix-list <*name*>

The **match ipv6 address prefix-list** command distributes any routes that have a destination IPv6 address permitted by a prefix list.

The <*name*> parameter specifies an IPv6 prefix list name.

Syntax: set ipv6 next-hop <*ipv6-address*>

The <*ipv6-address*> parameter specifies the IPv6 global address of the next-hop router. You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Configuring a BGP4+ peer group

If a has multiple neighbors with similar attributes, you can configure a peer group, then add neighbors to the group instead of configuring neighbors individually for all parameters as described in "Configuring BGP4+ neighbors using global or site-local IPv6 addresses" on page 1899 and "Adding BGP4+ neighbors using link-local addresses" on page 1900.

NOTE
You can add IPv6 neighbors only to an IPv6 peer group. You cannot add an IPv4 neighbor to an IPv6 peer group and vice versa. IPv6 and IPv6 peer groups must remain separate.

To configure a BGP4+ peer group, you must perform the tasks listed below.

1. Create a peer group.
2. Add a neighbor to the local .
3. Assign the IPv6 neighbor to the peer group.

Creating a BGP4+ peer group

To create a peer group named "peer_group1," enter the following commands:

```
NetIron(config-bgp)# address-family ipv6 unicast
NetIron(config-bgp-ipv6u)# neighbor peer_group1 peer-group
```

Syntax: neighbor <*peer-group-name*> **peer-group**

Specify a name for the peer group.

To delete the peer group, enter the **no** form of this command.

Adding a neighbor to a local

To add the IPv6 address 2001:efff:89::23 of a neighbor in remote AS 1001 to the BGP4+ neighbor table of a , enter the following command:

```
NetIron(config-bgp-ipv6u)# neighbor 2001:efff:89::23 remote-as 1001
```

NOTE
The example above adds an IPv6 neighbor at the BGP4+ unicast address family configuration level. This neighbor, by default, is enabled to exchange BGP4+ unicast prefixes. However, if you add an IPv6 neighbor while at the global BGP configuration or IPv4 BGP unicast address family configuration level, the neighbor will not exchange BGP4+ unicast prefixes until you explicitly enable it to do so by entering the **neighbor** <*ipv6-address*> | <*peer-group-name*> **activate** command at the BGP4+ unicast address family configuration level.

Syntax: neighbor <*ipv6-address*> **remote-as** <*as-number*>

The **ipv6-address** parameter specifies the IPv6 address of the neighbor. You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*as-number*> parameter indicates the number of the AS in which the neighbor resides.

To delete the neighbor from the BGP4+ neighbor table, enter the **no** form of this command.

Assigning IPv6 neighbor to peer group

To assign an already configured neighbor (2001:efff:89::23) to the peer group peer_group1, enter the following command at the BGP4+ unicast address family configuration level:

```
NetIron(config-bgp-ipv6u)# neighbor 2001:efff:89::23 peer-group peer_group1
```

Syntax: **neighbor** <*ipv6-address*> **peer-group** <*peer-group-name*>

The <*ipv6-address*> parameter specifies the IPv6 address of the neighbor. You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **peer-group** <*peer-group-name*> parameter indicates the name of the already created peer group.

To delete the mapping of the neighbor IPv6 address to the peer group, enter the **no** form of this command.

Advertising the default BGP4+ route

By default, the BGP4+ router does not originate and advertise a default BGP4+ route. A default route is the IPv6 address :: and the route prefix 0; that is, ::/0.

You can enable the BGP4+ router to advertise the default BGP4+ route by specifying the **default-information-originate** command at the BGP4+ unicast address family configuration level. Before entering this command, the default route ::/0 must be present in the IPv6 route table.

To enable the BGP4+ router to advertise the default route, enter the following command:

```
NetIron(config-bgp-ipv6u)# default-information-originate
```

Syntax: **[no] default-information-originate**

You can also enable the BGP4+ router to send the default route to a particular neighbor by specifying the **neighbor** <*ipv6-address*> **default-originate** command at the BGP4+ unicast address family configuration level. This command does not require the presence of the default route ::/0 in the IPv6 route table.

For example, to enable the BGP4+ router to send the default route to a neighbor with the IPv6 address of 2001:efff:89::23, enter a command such as the following:

```
NetIron(config-bgp-ipv6u)# neighbor 2001:efff:89::23 default-originate
```

Syntax: **[no] neighbor** <*ipv6-address*> **default-originate [route-map** <*name*>**]**

The <*ipv6-address*> parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Specifying the optional **route-map** <*name*> parameter injects the default route conditionally, based on the match conditions in the route map.

Importing routes into BGP4+

By default, the does not import routes into BGP4+. This section explains how to use the **network** command to enable the importing of specified routes into BGP4+.

NOTE
The routes imported into BGP4+ must first exist in the IPv6 unicast route table.

For example, to import the IPv6 prefix 3ff0:ec21::/32 into the BGP4+ database, enter the following command at the BGP4+ unicast address family configuration level:

```
NetIron(config-bgp-ipv6u)# network 3ff0:ec21::/32
```

Syntax: **network** <*ipv6-prefix*>/<*prefix-length*> [**route-map** <*name*>]

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

You can specify the optional **route-map** <*name*> parameter if you want to change attributes of a route when importing it into BGP4+.

To disable the importing of a specified route, enter the **no** form of this command without the route-map parameter.

Redistributing prefixes into BGP4+

You can configure the to redistribute routes from the following sources into BGP4+:

- Static IPv6 routes.
- Directly connected IPv6 networks.
- OSPFv3.
- RIPng.

You can redistribute routes in the following ways:

- By route types, for example, the redistributes all IPv6 static and RIPng routes.
- By using a route map to filter which routes to redistribute, for example, the redistributes specified IPv6 static and RIPng routes only.

For example, to configure the redistribution of all RIPng routes into the BGP4+ unicast database, enter the following commands at the BGP4+ address family configuration level:

```
NetIron(config-bgp-ipv6u)# redistribute rip
```

Syntax: **redistribute** <*protocol*> [**level-1 | level-1-2 | level-2**] [**match external1 | external2 | internal**] [**metric** <*metric-value*>] [**route-map** <*name*>]

The <*protocol*> parameter can be **connected**, **ospf**, **rip**, or **static**.

If you specify **ospf** as the protocol, you can optionally specify the redistribution of external 1, external 2, or internal routes. (The default is internal.)

The **metric** <*metric-value*> parameter specifies the metric used for the redistributed route. If a value is not specified for this option, and no value is specified using the **default-metric** command at the BGP4+ unicast address family configuration level, the metric value for the IPv6 static, RIPng, or IPv6 OSPF route is used. Use a value consistent with the destination protocol.

The <*name*> parameter specifies a route map name.

Aggregating routes advertised to BGP4 neighbors

By default, a advertises individual BGP4+ routes for all the networks. The aggregation feature allows you to configure a to aggregate routes in a range of networks into a single IPv6 prefix. For example, without aggregation, a will individually advertise routes for networks ff00:f000:0001:0000::/64, ff00:f000:0002:0000::/64,ff00:f000:0003:0000::/64, and so on. You can configure the to instead send a single, aggregate route for the networks. The aggregate route would be advertised as ff00:f000::/24 to BGP4 neighbors.

To aggregate BGP4+ routes for ff00:f000:0001:0000::/64, ff00:f000:0002:0000::/64,ff00:f000:0003:0000::/64, enter the following command.

```
NetIron(config-bgp)# aggregate-address ff00:f000::/24 summary-only
```

Syntax: **aggregate-address** <*ipv6-prefix*>/<*prefix-length*> **[as-set] [summary-only] [suppress-map** <*map-name*>**] [advertise-map** <*map-name*>**] [attribute-map** <*map-name*>**]**

The <*ipv6-prefix*>/<*prefix-length*> parameter specifies the aggregate value for the networks. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **as-set** keyword causes the device to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.

The **summary-only** keyword prevents the device from advertising more specific routes contained within the aggregate route.

The **suppress-map** <*map-name*> parameter prevents the more specific routes contained in the specified route map from being advertised.

The **advertise-map** <*map-name*> parameter configures the device to advertise the more specific routes in the specified route map.

The **attribute-map** <*map-name*> parameter configures the device to set attributes for the aggregate routes based on the specified route map.

NOTE
For the **suppress-map**, **advertise-map**, and **attribute-map** parameters, the route map must already be defined.

To remove an aggregate route from a BGP4 neighbor advertisement, use the **no** form of this command without any parameters.

Using route maps

You can use a route map to filter and change values in BGP4+ routes. Currently, you can apply a route map to IPv6 unicast routes that are independent of IPv4 routes.

To configure a route map to match on IPv6 unicast routes, enter commands such as the following.

```
NetIron(config)# router bgp
NetIron(config-bgp)# address-family ipv6 unicast
NetIron(config-bgp-ipv6u)# neighbor 2001:eff3:df78::67 remote-as 1001
NetIron(config-bgp-ipv6u)# neighbor 2001:eff3:df78::67 route-map in map1
```

```
NetIron(config-bgp-ipv6u)# exit
NetIron(config)# ipv6 prefix-list ipv6_uni seq 10 permit 2001:eff3::/32
NetIron(config)# route-map map1 permit 10
NetIron(config-routemap-map1)# match ipv6 address prefix-list ipv6_uni
```

This example configures a route map named “map1” that permits incoming IPv6 unicast routes that match the prefix list named “ipv6_uni” (2001:eff3::/32). Note that you apply the route map while at the BGP4+ unicast address family configuration level.

Clearing BGP4+ information

This section contains information about clearing the following for BGP4+:

- Route flap dampening.
- Route flap dampening statistics.
- Neighbor information.
- BGP4+ routes in the IPv6 route table.
- Neighbor traffic counters.

NOTE
The **clear** commands implemented for BGP4+ correspond to the **clear** commands implemented for IPv4 BGP. For example, you can specify the **clear ipv6 bgp flap-statistics** command for IPv6 and the **clear ip bgp flap-statistics** for IPv4.

Removing route flap dampening

You can un-suppress routes by removing route flap dampening from the routes. The allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 bgp dampening
```

Syntax: clear ipv6 bgp dampening [<*ipv6-prefix*>/<*prefix-length*>]

You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

To un-suppress a specific route, enter a command such as the following.

```
NetIron# clear ipv6 bgp dampening 2001:e0ff::/32
```

This command un-suppresses only the routes for network 2001:e0ff::/32.

Clearing route flap dampening statistics

The allows you to clear all route flap dampening statistics or statistics for a specified IPv6 prefix or a regular expression.

NOTE
Clearing the dampening statistics for a route does not change the dampening status of the route.

To clear all the route dampening statistics, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 bgp flap-statistics
```

Syntax: **clear ipv6 bgp flap-statistics** [<*ipv6-prefix*>/<*prefix-length*> | **neighbor** <*ipv6-address*> | **regular-expression** <*regular-expression*>]

The <*ipv6-prefix*>/<*prefix-length*> parameter clears route flap dampening statistics for a specified IPv6 prefix. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **neighbor** <*ipv6-address*> parameter clears route flap dampening statistics only for routes learned from the neighbor with the specified IPv6 address.

The **regular-expression** <*regular-expression*> parameter is a regular expression.

Clearing BGP4+ local route information

You can clear locally imported or routes redistributed into BGP4+.

To clear all local route information, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 bgp local routes
```

Syntax: **clear ipv6 bgp local routes**

Clearing BGP4+ neighbor information

You can perform the following tasks related to BGP4+ neighbor information:

- Clear diagnostic buffers.
- Reset a session to send and receive Outbound Route Filters (ORFs).
- Close a session, or reset a session and resend or receive an update.
- Clear traffic counters.
- Clear route flap dampening statistics.

Clearing BGP4+ neighbor diagnostic buffers

You can clear the following BGP4+ neighbor diagnostic information in buffers:

- The first 400 bytes of the last packet that contained an error.
- The last NOTIFICATION message either sent or received by the neighbor.

To display these buffers, use the **last-packet-with-error** keyword with the **show ipv6 bgp neighbors** command. For more information about this command, refer to "Displaying last error packet from a BGP4+ neighbor" on page 1937.

You can clear the buffers for all neighbors, for an individual neighbor, or for all the neighbors within a specific peer group or AS.

To clear these buffers for neighbor 2000:e0ff:37::1, enter the following commands at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 bgp neighbor 2000:e0ff:37::1 last-packet-with-error
NetIron# clear ipv6 bgp neighbor 2000:e0ff:37::1 notification-errors
```

Syntax: clear ipv6 bgp neighbor all | <*ipv6-address*> **|** <*peer-group-name*> **|** <*as-number*> **last-packet-with-error | notification-errors**

The **all** | <*ipv6-address*> | <*peer-group-name*> | <*as-num*> specifies the neighbor. The <*ipv6-address*> parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373. The <*peer-group-name*> specifies all neighbors in a specific peer group. The <*as-num*> parameter specifies all neighbors within the specified AS. The **all** keyword specifies all neighbors.

The **last-packet-with-error** keyword clears the buffer containing the first 400 bytes of the last packet that contained errors.

The **notification-errors** keyword clears the notification error code for the last NOTIFICATION message sent or received.

Resetting a BGP4+ neighbor session to send and receive ORFs

You can perform a hard or soft reset of a BGP4+ neighbor session to send or receive ORFs.

To perform a hard reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
NetIron# clear ipv6 bgp neighbor 2000:e0ff:38::1
```

This command resets the BGP4+ session with neighbor 2000:e0ff:38::1 and sends the ORFs to the neighbor when the neighbor comes up again. If the neighbor sends ORFs to the , the accepts them if the send capability is enabled.

To perform a soft reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
NetIron(config)# clear ipv6 bgp neighbor peer_group1 soft in prefix-list
```

Syntax: clear ipv6 bgp neighbor <*ipv6-address*> **|** <*peer-group-name*> **[soft in prefix-filter]**

The <*ipv6-address*> parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*peer-group-name*> specifies all neighbors in a specific peer group.

If you use the **soft in prefix-filter** keyword, the sends an updated IPv6 prefix list to the neighbor as part of its route refresh message to the neighbor.

Closing or resetting a BGP4+ neighbor session

You can close a neighbor session or resend route updates to a neighbor. You can specify all neighbors, a single neighbor, or all neighbors within a specific peer group or AS.

If you close a neighbor session, the and the neighbor clear all the routes they learned from each other. When the and neighbor establish a new BGP4+ session, they exchange route tables again. Use this method if you want the to relearn routes from the neighbor and resend its own route table to the neighbor.

If you use the **soft-outbound** keyword, the compiles a list of all the routes it would normally send to the neighbor at the beginning of a session. However, before sending the updates, the also applies the filters and route maps you have configured to the list of routes. If the filters or route maps result in changes to the list of routes, the sends updates to advertise, change, or even withdraw routes on the neighbor as needed. This ensures that the neighbor receives only the routes you want it to contain. Even if the neighbor already contains a route learned from the that you later decided to filter out, using the soft-outbound option removes that route from the neighbor. If no change is detected from the previously sent routes, an update is not sent.

For example, to close all neighbor sessions and thus flush all the routes exchanged by the and all neighbors, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 bgp neighbor all
```

Syntax: clear ipv6 bgp neighbor all | <*ipv6-address*> **|** <*peer-group-name*> **|** <*as-number*> **[soft-outbound | soft [in | out]]**

The **all** | <*ipv6-address*> | <*peer-group-name*> | <*as-number*> specifies the neighbor. The <*ipv6-address*> parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373. The <*peer-group-name*> specifies all neighbors in a specific peer group. The <*as-number*> parameter specifies all neighbors within the specified AS. The **all** keyword specifies all neighbors.

Use the **soft-outbound** keyword to perform a soft reset of a neighbor session and resend only route update changes to a neighbor.

Use the **soft in** parameter to perform a soft reset of a neighbor session and requests a route update from a neighbor.

Use the **soft out** parameter to perform a soft reset of a neighbor session and resend all routes to a neighbor.

Clearing BGP4+ neighbor traffic counters

You can clear the BGP4+ message counter (reset them to 0) for all neighbors, a single neighbor, or all neighbors within a specific peer group or AS.

For example, to clear the BGP4+ message counter for all neighbors within an AS 1001, enter a command such as the following at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 bgp neighbor 1001 traffic
```

Syntax: clear ipv6 bgp neighbor all | <*ipv6-address*> **|** <*peer-group-name*> **|** <*as-number*> **traffic**

The **all** | <*ipv6-address*> | <*peer-group-name*> | <*as-number*> specifies the neighbor. The <*ipv6-address*> parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373. The <*peer-group-name*> specifies all neighbors in a specific peer group. The <*as-number*> parameter specifies all neighbors within the specified AS. The **all** keyword specifies all neighbors.

Specify the **traffic** keyword to clear the BGP4+ message counter.

Clearing BGP4+ neighbor route flap dampening statistics

The allows you to clear all route flap dampening statistics for a specified BGP4+ neighbor.

NOTE
Clearing the dampening statistics for a neighbor does not change the dampening status of a route.

To clear all of the route flap dampening statistics for a neighbor, enter a command such as the following at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 bgp neighbor 2000:e0ff:47::1 flap-statistics
```

Syntax: clear ipv6 bgp neighbor <*ipv6-address*> **flap-statistics**

The <*ipv6-address*> parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Specify the **flap-statistics** keyword to clear route flap dampening statistics for the specified neighbor.

Clearing and resetting BGP4+ routes in the IPv6 route table

You can clear all BGP4+ routes or only those routes associated with a particular IPv6 prefix from the IPv6 route table and reset the routes. When cleared, the BGP4+ routes are removed from the IPv6 main route table and then restored again.

For example, to clear all BGP4+ routes and reset them, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
NetIron# clear ipv6 bgp routes
```

Syntax: clear ip bgp routes [<*ipv6-prefix*>/<*prefix-length*>]

The <*ipv6-prefix*>/<*prefix-length*> parameter clears routes associated with a particular IPv6 prefix. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

Clearing traffic counters for all BGP4+ neighbors

To clear the message counters (reset them to 0) for all BGP4+ neighbors, enter the following command.

```
NetIron(config)# clear ipv6 bgp traffic
```

Syntax: clear ipv6 bgp traffic

Displaying BGP4+ information

You can display the following BGP4+ information:

- BGP4+ route table.
- BGP4+ route information.
- BGP4+ route-attribute entries.

- BGP4+ configuration information.
- Dampened BGP4+ paths.
- Filtered-out BGP4+ routes.
- BGP4+ route flap dampening statistics.
- BGP4+ neighbor information.
- BGP4+ peer group configuration information.
- BGP4+ summary information.

NOTE
The **show** commands implemented for BGP4+ correspond to the **show** commands implemented for IPv4 BGP. For example, you can specify the **show ipv6 bgp** command for IPv6 and the **show ip bgp** command for IPv4. Also, the displays for the IPv4 and IPv6 versions of the **show** commands are similar except where relevant, IPv6 neighbor addresses replace IPv4 neighbor addresses, IPv6 prefixes replace IPv4 prefixes, and IPv6 next-hop addresses replace IPv4 next-hop addresses.

Displaying the BGP4+ route table

BGP4+ uses filters you define, as well as an algorithm to determine the preferred route to a destination. (For information about the algorithm, see .) BGP4+ sends only the preferred route to the 's IPv6 table. However, if you want to view all the routes BGP4+ knows about, you can display the BGP4+ table.

To display the BGP4+ route table, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp routes
Total number of BGP Routes: 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix              Next Hop        Metric     LocPrf     Weight Status
1      2002::/16           ::              1          100        32768  BL
         AS_PATH:
2      2002:1234::/32      ::              1          100        32768  BL
         AS_PATH:
```

This display shows the following information.

TABLE 347 Summary of BGP4+ routes

This field...	Displays...
Number of BGP4+ Routes	The number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The route's prefix.
Next Hop	The next-hop router for reaching the route from the .
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the advertised route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.

TABLE 347 Summary of BGP4+ routes (Continued)

This field...	Displays...
Weight	The value that this associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Status	The route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4+ has determined that this is the optimal route to the destination. • b – NOT-INSTALLED-BEST – BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • E – EBGP. The route was learned through a in another AS. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – IBGP. The route was learned through a in the same AS. • L – LOCAL. The route originated on this . • M – MULTIPATH. BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
AS-PATH	The AS-path information for the route.

Syntax: **show ipv6 bgp routes** [<*ipv6-prefix*>/<*prefix-length*> | <*table-entry-number*> | **age** <*seconds*> | **as-path-access-list** <*name*> | **as-path-filter** <*number*> | **best** | **cidr-only** | [**community** <*number*> | **no-export** | **no-advertise** | **internet** | **local-as**] | **community-access-list** <*name*> | **community-filter** <*number*> | **detail** [<*option*>] | **local** | **neighbor** <*ipv6-address*> | **nexthop** <*ipv6-address*> | **no-best** | **prefix-list** <*name*> | **regular-expression** <*regular-expression*> | **route-map** <*name*> | **summary** | **unreachable**]

You can use the following options with the **show ipv6 bgp routes** command to determine the content of the display:

The <*ipv6-prefix*>/<*prefix-length*> parameter displays routes for a specific network. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The <*table-entry-number*> parameter specifies the table entry with which you want the display to start. For example, if you specify 100, the display shows entry 100 and all entries subsequent to entry 100.

The **age** <*seconds*> parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** <*name*> parameter filters the display using the specified AS-path ACL.

The **as-path-filter** <*number*> parameter filters the display using the specified AS-path filter.

The **best** keyword displays the routes received from neighbors that the selected as the best routes to their destinations.

The **cidr-only** keyword lists only the routes whose network masks do not match their class network length.

The **community** <*number*> parameter lets you display routes for a specific community. You can specify **local-as**, **no-export**, **no-advertise**, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of up to 1– 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** <*name*> parameter filters the display using the specified community ACL.

The **community-filter** <*number*> parameter lets you display routes that match a specific community filter.

The **detail** <*option*> parameter lets you display more details about the routes. You can refine your request by also specifying one of the other parameters after the **detail** keyword.

The **local** keyword displays routes that are local to the .

The **neighbor** <*ipv6-address*> parameter displays routes learned from a specified BGP4+ neighbor.

The **nexthop** <*ipv6-address*> parameter displays the routes for a specified next-hop IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **no-best** keyword displays the routes for which none of the routes to a given prefix were selected as the best route. The IPv6 route table does not contain a BGP4+ route for any of the routes listed using this option.

The **prefix-list** <*name*> parameter filters the display using the specified IPv6 prefix list.

The **regular-expression** <*regular-expression*> parameter filters the display based on a regular expression.

The **route-map** <*name*> parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The **summary** keyword displays summary information for the routes.

The **unreachable** keyword displays the routes that are unreachable because the does not have a valid RIPng, OSPFv3, or static IPv6 route to the next hop.

To display details about BGP4+ routes, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp routes detail
Total number of BGP Routes: 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
1       Prefix: 2002::/16,  Status: BL,  Age: 2d17h10m42s
         NEXT_HOP: ::,  Learned from Peer: Local Router
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:
            Adj_RIB_out count: 1,  Admin distance 190
2       Prefix: 2002:1234::/32,  Status: BL,  Age: 2d17h10m42s
         NEXT_HOP: ::,  Learned from Peer: Local Router
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:
            Adj_RIB_out count: 1,  Admin distance 190
```

This display shows the following information.

TABLE 348 Detailed BGP4+ route information

This field...	Displays...
Number of BGP4+ Routes advertised to specified neighbor (appears only in display for all routes)	For information about this field, refer to Table 347 on page 1911.
Status codes	For information about this field, refer to Table 347 on page 1911.
Prefix	For information about this field, refer to Table 347 on page 1911.
Status	For information about this field, refer to Table 347 on page 1911.
Age	The age of the advertised route, in seconds.
Next Hop	For information about this field, refer to Table 347 on page 1911.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. “Local Router” indicates that the itself learned the route.
LOCAL_PREF	For information about this field, refer to Table 347 on page 1911.
MED	The value of the advertised route’s MED attribute. If the route does not have a metric, this field is blank.

TABLE 348 Detailed BGP4+ route information (Continued)

This field...	Displays...
Origin	The source of the route information. The origin can be one of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4+ has determined that this is the optimal route to the destination. • b – NOT-INSTALLED-BEST – BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • EGP – The routes with this set of attributes came to BGP4+ through EGP. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • IGP – The routes with this set of attributes came to BGP4+ through IGP. • L – LOCAL. The route originated on this . • M – MULTIPATH. BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
Weight	For information about this field, refer to Table 347 on page 1911.
AS-PATH	For information about this field, refer to Table 347 on page 1911.
Adj_RIB_out count	The number of neighbors to which the route has been or will be advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4+ neighbor.
Admin Distance	The administrative distance of the route.

Syntax: **show ipv6 bgp routes detail** [<*ipv6-prefix*>/<*prefix-length*> | <*table-entry-number*> | **age** <*seconds*> | **as-path-access-list** <*name*> | **as-path-filter** <*number*> | **best** | **cidr-only** | [**community** <*number*> | **no-export** | **no-advertise** | **internet** | **local-as**] | **community-access-list** <*name*> | **community-filter** <*number*> | **local** | **neighbor** <*ipv6-address*> | **nexthop** <*ipv6-address*> | **no-best** | **prefix-list** <*name*> | **regular-expression** <*regular-expression*> | **route-map** <*name*> | **summary** | **unreachable**]

You can use the following options with the **show ipv6 bgp routes detail** command to determine the content of the display.

The <*ipv6-prefix*>/<*prefix-length*> option displays details about routes for a specific network. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The <*table-entry-number*> parameter specifies the table entry with which you want the display to start. For example, if you specify 100, the display shows entry 100 and all entries subsequent to entry 100.

The **age** <*seconds*> parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** <*name*> parameter filters the display using the specified AS-path ACL.

The **as-path-filter** <*number*> parameter filters the display using the specified AS-path filter.

The **best** keyword displays the routes received from neighbors that the selected as the best routes to their destinations.

The **cidr-only** keyword lists only the routes whose network masks do not match their class network length.

The **community** <*number*> parameter lets you display routes for a specific community. You can specify **local-as**, **no-export**, **no-advertise**, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of up to 1– 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** <*name*> parameter filters the display using the specified community ACL.

The **community-filter** <*number*> parameter lets you display routes that match a specific community filter.

The **detail** keyword lets you display more details about the routes. You can refine your request by also specifying one of the other parameters after the **detail** keyword.

The **local** keyword displays routes that are local to the .

The **neighbor** <*ipv6-address*> parameter displays routes learned from a specified BGP4+ neighbor.

The **nexthop** <*ipv6-address*> option displays the routes for a specified next-hop IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **no-best** keyword displays the routes for which none of the routes to a given prefix were selected as the best route. The IPv6 route table does not contain a BGP4+ route for any of the routes listed using this option.

The **prefix-list** <*name*> parameter filters the display using the specified IPv6 prefix list.

The **regular-expression** <*regular-expression*> parameter filters the display based on a regular expression.

The **route-map** <*name*> parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The **summary** keyword displays summary information for the routes.

The **unreachable** keyword displays the routes that are unreachable because the does not have a valid RIPng, OSPFv3 or static IPv6 route to the next hop.

Displaying BGP4+ route information

You can display all BGP4+ routes known by a , only those routes that match a specified prefix, or routes that match a specified or longer prefix.

To display all BGP4+ routes known by the , enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp
Total number of BGP Routes: 2
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  2002::/16          ::              1      100    32768  ?
*>  2002:1234::/32     ::              1      100    32768  ?
```

Syntax: show ipv6 bgp <*ipv6-prefix*>/<*prefix-length*> **[longer-prefixes]**

The <*ipv6-prefix*>/<*prefix-length*> parameter allows you to display routes that match a specified BGP prefix only. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **longer-prefixes** keyword allows you to display routes that match a specified or longer BGP prefix. For example, if you specify **2002::/16 longer-prefixes**, then all routes with the prefix 2002::/16 or that have a longer prefix (such as 2002:e016::/32) are displayed.

To display only those routes that match prefix 2002::/16, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp 2002::/16
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  2002::/16          ::              1      100    32768  ?
       Route is advertised to 1 peers:
        2000:4::110(65002)
```

For example, to display routes that match prefix 2002::/16 or longer, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp 2002::/16 longer-prefixes
Number of BGP Routes matching display condition : 3
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  2002::/16          ::              1      100    32768  ?
*>  2002:1234::/32     ::              1      100    32768  ?
*>  2002:e0ff::/32     ::              1      100    32768  ?
       Route is advertised to 1 peers:
        2000:4::110(65002)
```

These displays show the following information.

TABLE 349 BGP4+ route information

This field...	Displays...
Total number of BGP Routes (appears in display of all BGP routes only)	The number of routes known by the .
Number of BGP Routes matching display condition (appears in display that matches specified and longer prefixes)	The number of routes that matched the display parameters you entered. This is the number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output.
Origin codes	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.
Network	The network prefix and prefix length.
Next Hop	The next-hop router for reaching the network from the .
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
Weight	The value that this associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Path	The route's AS path.

Displaying BGP4+ route-attribute entries

The route-attribute entries table lists sets of BGP4+ attributes stored in the 's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the typically has fewer route attribute entries than routes.

To display the IPv6 route-attribute entries table, enter the following command.

```
NetIron# show ipv6 bgp attribute-entries
        Total number of BGP Attribute Entries: 378
1       Next Hop  :::                 Metric   :1              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :(65002) 65001 4355 2548 3561 5400 6669 5548
        Address: 0x27a4cdb0  Hash:877 (0x03000000) Reference Counts: 2:0:2
...
```

NOTE
Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

Syntax: show ipv6 bgp attribute-entries

For information about display displaying route-attribute entries for a specified BGP4+ neighbor, refer to "Displaying BGP4+ neighbor route-attribute entries" on page 1935.

This display shows the following information.

TABLE 350 BGP4+ route-attribute entries information

This field...	Displays...
Total number of BGP Attribute Entries	The number of entries contained in the 's BGP4+ route-attribute entries table.
Next Hop	The IPv6 address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP4+ through EGP. • IGP – The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP, and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route-reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • AS Number shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • Router-ID shows the router that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred • None – Indicates this attribute is not present. **NOTE:** Information loss under these circumstances is a normal part of BGP4+ and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	For debugging purposes only.
Hash	For debugging purposes only.
Reference Counts	For debugging purposes only.

Displaying the BGP4+ running configuration

To view the active BGP4+ configuration information contained in the running configuration without displaying the entire running configuration, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp config
Current BGP configuration:
router bgp
 local-as 1000
 neighbor peer_group1 peer-group
 neighbor 2001:4383:e0ff:783a::3 remote-as 1001
 neighbor 2001:4484:edd3:8389::1 remote-as 1002
 neighbor 2001:efff:80::23 peer-group peer_group1
 neighbor 2001:efff:80::23 remote-as 1003
 address-family ipv4 unicast
 no neighbor 2001:4383:e0ff:783a::3 activate
 no neighbor 2001:4484:edd3:8389::1 activate
 no neighbor 2001:efff:80::23 activate
 exit-address-family

 address-family ipv4 multicast
 exit-address-family

 address-family ipv6 unicast
 network 3ff0:ec21::/32
 neighbor peer_group1 activate
 neighbor 2001:4484:edd3:8389::1 activate
 exit-address-family

end
```

Syntax: show ipv6 bgp config

Displaying dampened BGP4+ paths

To display BGP4+ paths that have been dampened (suppressed) by route flap dampening, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp dampened-paths
Status Code >:best d:damped h:history *:valid
    Network From Flaps Since Reuse Path
*d  8::/13 2000:1:1::1 1 0 :1 :14 0 :2 :20 100 1002 1000
*d  1::/16 2000:1:1::1 1 0 :1 :14 0 :2 :20 100 1002 1000
*d  4::/14 2000:1:1::1 1 0 :1 :14 0 :2 :20 100 1002 1000
*d  2::/15 2000:1:1::1 1 0 :1 :14 0 :2 :20 100 1002 1000
*d  0:8000::/17 2000:1:1::1 1 0 :1 :14 0 :2 :20 100 1002 1000
*d  2000:1:17::/642000:1:1::1 1 0 :1 :18 0 :2 :20 100
```

Syntax: show ipv6 bgp dampened-paths

This display shows the following information.

TABLE 351 Dampened BGP4+ path information

This field...	Displays...
Status codes	A list of the characters the display uses to indicate the path's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output. The status column displays a "d" for each dampened route.
Network	The destination network of the route.
From	The IPv6 address of the advertising peer.
Flaps	The number of times the path has flapped.
Since	The amount of time (in hh:mm:ss) since the first flap of this route.
Reuse	The amount of time (in hh:mm:ss) after which the path is available again.
Path	The AS path of the route.

Displaying filtered-out BGP4+ routes

When you enable the soft reconfiguration feature, the saves all updates received from the specified neighbor or peer group. The saved updates include those that contain routes that are filtered out by the BGP4+ route policies in effect on the .

You can display a summary or more detailed information about routes that have been filtered out by BGP4+ route policies.

To display a summary of the routes that have been filtered out by BGP4+ route policies, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp filtered-routes
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix              Next Hop        Metric     LocPrf     Weight Status
1      3000::/16           2000:4::110                 100        0        EF
         AS_PATH: 65001 4355 701 80
2      4000::/16           2000:4::110                 100        0        EF
         AS_PATH: 65001 4355 1
3      5000::/16           2000:4::110                 100        0        EF
         AS_PATH: 65001 4355 701 1 189
```

The routes displayed by the command are the routes that the 's BGP policies filtered out. The did not place the routes in the BGP4+ route table, but did keep the updates. If a policy change causes these routes to be permitted, the does not need to request the route information from the neighbor, but instead uses the information in the updates.

Syntax: show ipv6 bgp filtered-routes [<*ipv6-prefix*>/<*prefix-length*> **[longer-prefixes] | [as-path-access-list** <*name*>**] | [prefix-list** <*name*>**]**

The <*ipv6-prefix*>/<*prefix-length*> parameter displays the specified IPv6 prefix of the destination network only. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **longer-prefixes** keyword allows you to display routes that match a specified or longer IPv6 prefix. For example, if you specify **2002::/16 longer-prefixes**, then all routes with the prefix 2002::/16 or that have a longer prefix (such as 2002:e016::/32) are displayed.

The **as-path-access-list** <*name*> parameter specifies an AS-path ACL. Specify an ACL name. Only the routes permitted by the AS-path ACL are displayed.

The **prefix-list** <*name*> parameter specifies an IPv6 prefix list. Only the routes permitted by the prefix list are displayed.

This display shows the following information.

TABLE 352 Summary of filtered-out BGP4+ route information

This field...	Displays...
Number of BGP4+ Routes matching display condition	The number of routes that matched the display parameters you entered. This is the number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output. The status column displays an "F" for each filtered route.
Prefix	The network address and prefix.
Next Hop	The next-hop router for reaching the network from the .
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.

TABLE 352 Summary of filtered-out BGP4+ route information (Continued)

This field...	Displays...
Weight	The value that this associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Status	The route's status, which can be one or more of the following: • A – AGGREGATE – The route is an aggregate route for multiple networks. • B – BEST – BGP4+ has determined that this is the optimal route to the destination. • b – NOT-INSTALLED-BEST – BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • C – CONFED_EBGP – The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED – This route has been dampened (by the route dampening feature), and is currently unusable. • E – EBGP – The route was learned through a in another AS. • H – HISTORY – Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – IBGP – The route was learned through a in the same AS. • L – LOCAL – The route originated on this . • M – MULTIPATH – BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S – SUPPRESSED – This route was suppressed during aggregation and thus is not advertised to neighbors. • F – FILTERED – This route was filtered out by BGP4+ route policies on the , but the saved updates containing the filtered routes.

To display detailed information about the routes that have been filtered out by BGP4+ route policies, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp filtered-routes detail
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
1   Prefix: 800:2:1::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
2   Prefix: 900:1:18::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
3   Prefix: 1000:1:1::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
4   Prefix: 2000:1:1::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
5   Prefix: 2000:1:11::1/128, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: igp, Weight: 0
    AS_PATH: 100
6   Prefix: 2000:1:17::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
```

Syntax: show ipv6 bgp filtered-routes detail [<*ipv6-prefix*>/<*prefix-length*> **[longer-prefixes] | [as-path-access-list** <*name*>] **| [prefix-list** <*name*>]

The <*ipv6-prefix*>/<*prefix-length*> parameter displays the specified IPv6 prefix of the destination network only. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **longer-prefixes** keyword allows you to display routes that match a specified or longer IPv6 prefix. For example, if you specify **2002::/16 longer-prefixes**, then all routes with the prefix 2002::/16 or that have a longer prefix (such as 2002:e016::/32) are displayed.

The **as-path-access-list** <*name*> parameter specifies an AS-path ACL. Only the routes permitted by the AS-path ACL are displayed.

The **prefix-list** <*name*> parameter specifies an IPv6 prefix list. Only the routes permitted by the prefix list are displayed.

This display shows the following information.

TABLE 353 Detailed filtered-out BGP4+ route information

This field...	Displays...
Status codes	A list of the characters the display uses to indicate the route's status. The Status field display an "F" for each filtered route.
Prefix	For information about this field, refer to Table 352 on page 1922.

TABLE 353 Detailed filtered-out BGP4+ route information (Continued)

This field...	Displays...
Status	For information about this field, refer to Table 352 on page 1922.
Age	The age of the route, in seconds.
Next hop	For information about this field, refer to Table 352 on page 1922.
Learned from peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the itself learned the route.
Local pref	For information about this field, refer to Table 352 on page 1922.
MED	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • A – AGGREGATE – The route is an aggregate route for multiple networks. • B – BEST – BGP4+ has determined that this is the optimal route to the destination. • b – NOT-INSTALLED-BEST – BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • C – CONFED_EBGP – The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED – This route has been dampened (by the route dampening feature), and is currently unusable. • E – EBGP – The route was learned through a in another AS. • H – HISTORY – Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – IBGP – The route was learned through a in the same AS. • L – LOCAL – The route originated on this . • M – MULTIPATH – BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S – SUPPRESSED – This route was suppressed during aggregation and thus is not advertised to neighbors. • F – FILTERED – This route was filtered out by BGP4+ route policies on the , but the saved updates containing the filtered routes.
Weight	For information about this field, refer to Table 352 on page 1922.
AS path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.

Displaying route flap dampening statistics

To display route dampening statistics for all dampened routes, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp flap-statistics
Total number of flapping routes: 14
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flaps Since    Reuse    Path
h>  2001:2::/32        3001:23::47      1    0 :0 :13 0 :0 :0  65001 4355 1 701
*>  3892:34::/32       3001:23::47      1    0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: **show ipv6 bgp flap-statistics** [<*ipv6-prefix*>/<*prefix-length*> **[longer-prefixes]** | **as-path-filter** <*number*> | **neighbor** <*ipv6-address*> | **regular-expression** <*regular-expression*>]

The <*ipv6-prefix*>/<*prefix-length*> parameter displays statistics for the specified IPv6 prefix only. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **longer-prefixes** keyword allows you to display statistics for routes that match a specified or longer IPv6 prefix. For example, if you specify **2000::/16 longer-prefixes**, then all routes with the prefix 2002:: or that have a longer prefix (such as 2002:e016::/32) are displayed.

The **as-path-filter** <*number*> parameter specifies an AS path filter to display. Specify a filter number.

The **neighbor** <*ipv6-address*> parameter displays statistics for routes learned from the specified neighbor only. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ipv6 bgp neighbor** <*ipv6-address*> **flap-statistics**.

The **regular-expression** <*regular-expression*> parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters.

You can also display route flap dampening statistics for a specified IPv6 neighbor. For more information, refer to "Displaying route flap dampening statistics for a BGP4+ neighbor" on page 1936.

This display shows the following information.

TABLE 354 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the 's BGP4+ route table that have changed state and thus have been marked as flapping routes.
Status code	Indicates the dampening status of the route, which can be one of the following: • > – This is the best route among those in the BGP4+ route table to the route's destination. • d – This route is currently dampened, and thus unusable. • h – The route has a history of flapping and is unreachable now. • * – The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The IPv6 address of the advertising peer.
Flaps	The number of flaps (state changes) the route has experienced.

TABLE 354 Route flap dampening statistics

This field...	Displays...
Since	The amount of time (in hh:mm:ss) since the first flap of this route.
Reuse	The amount of time (in hh:mm:ss) after which the path is again available.
Path	The AS path of the route.

You also can display all the dampened routes by using the **show ipv6 bgp dampened-paths** command. For more information, refer to "Displaying dampened BGP4+ paths" on page 1920.

Displaying BGP4+ neighbor information

You can display the following information about a 's BGP4+ neighbors:

Configuration information and statistics:

- Router advertisements.
- Route-attribute entries.
- Route flap dampening statistics.
- The last packet containing an error.
- Received Outbound Route Filters (ORFs).
- Routes received from a neighbor.
- BGP4+ Routing Information Base (RIB).
- Received best, not installed best, and unreachable routes.
- Route summary.

Displaying IPv6 neighbor configuration information and statistics

To display BGP4+ neighbor configuration information and statistics, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2000:4::110
1   IP Address: 2000:4::110, AS: 65002 (EBGP), RouterID: 1.1.1.1
    State: ESTABLISHED, Time: 5d20h38m54s, KeepAliveTime: 60, HoldTime: 180
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       2       8012      0            0
       Received: 1       0       7880      0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV6 unicast  capability
      Peer configured for IPV6 unicast  Routes
    TCP Connection state: ESTABLISHED
       Byte Sent:   152411, Received: 149765
       Local host:  2000:4::106, Local  Port: 8222
       Remote host: 2000:4::110, Remote Port: 179
       ISentSeq:  740437769  SendNext:  740590181  TotUnAck:          0
       TotSent:      152412  ReTrans:           0  UnAckSeq:  740590181
       IRcvSeq:   242365900  RcvNext:   242515666  SendWnd:       16384
       TotalRcv:     149766  DupliRcv:          0  RcvWnd:        16384
       SendQue:           0  RcvQue:            0  CngstWnd:       1440
...
```

NOTE
Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

In this example, the number in the far left column indicates the neighbor for which information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the 's Transmission Control Block (TCB) for the TCP session between the and its neighbor. These fields are described in detail in section 3.2 of RFC 793, "Transmission Control Protocol Functional Specification".

Syntax: show ipv6 bgp neighbor [<*ipv6-address*>]

The <*ipv6-address*> parameter allows you to display information for a specified neighbor only. You must specify the <*ipv6-address*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

This display shows the following information.

TABLE 355 BGP4+ neighbor configuration information and statistics

This field...	Displays...
IP Address	The IPv6 address of the neighbor.
AS	The AS in which the neighbor resides.
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP – The neighbor is in another AS. • EBGP_Confed – The neighbor is a member of another sub-AS in the same confederation. • IBGP – The neighbor is in the same AS.
RouterID	The neighbor's router ID.

State	The state of the 's session with the neighbor. The states are from the 's perspective of the session, not the neighbor's perspective. The state values can be one of the following:
	• IDLE – The BGP4+ process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4+ process.
	  • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• ADMND – The neighbor has been administratively shut down.
	  • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• CONNECT – BGP4+ is waiting for the connection process for the TCP neighbor session to be completed.
	• ACTIVE – BGP4+ is waiting for a TCP connection from the neighbor.
	NOTE: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.
	• OPEN SENT – BGP4+ is waiting for an Open message from the neighbor.
	• OPEN CONFIRM – BGP4+4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle.
	• ESTABLISHED – BGP4+ is ready to exchange UPDATE messages with the neighbor.
	  • If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed.
	NOTE: If you display information for the neighbor using the **show ipv6 bgp neighbor** <*ipv6-address*> command, the TCP receiver queue value will be greater than 0.

Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this sends keep alive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the will wait for a KEEPALIVE or UPDATE message from a BGP4+ neighbor before deciding that the neighbor is dead.
RefreshCapability	Whether the has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
Messages Sent and Received	The number of messages this has sent to and received from the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req

TABLE 355 BGP4+ neighbor configuration information and statistics (Continued)

This field...	Displays...
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following:
	• No abnormal error has occurred.
	• Reasons described in the BGP specifications:
	  • Message Header Error
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • OPEN Message Error
	  • Unsupported Version Number
	  • Bad Peer AS Number
	  • Bad BGP Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unsupported Capability
	  • UPDATE Message Error
	  • Malformed Attribute List
	  • Unrecognized Well-known Attribute
	  • Missing Well-known Attribute
	  • Attribute Flags Error
	  • Attribute Length Error
	  • Invalid ORIGIN Attribute
	  • Invalid NEXT_HOP Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS_PATH
	  • Hold Timer Expired
	  • Finite State Machine Error
	  • Rcv Notification

Last Connection Reset Reason (cont.)	• Reasons specific to the implementation:
	  • Reset All Peer Sessions
	  • User Reset Peer Session
	  • Port State Down
	  • Peer Removed
	  • Peer Shutdown
	  • Peer AS Number Change
	  • Peer AS Confederation Change
	  • TCP Connection KeepAlive Timeout
	  • TCP Connection Closed by Remote
	  • TCP Data Stream Error Detected

TABLE 355 BGP4+ neighbor configuration information and statistics (Continued)

This field...	Displays...

Notification Sent	If the receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.
	• Message Header Error
	  • Connection Not Synchronized
	  • Bad Message Length
	  • Bad Message Type
	  • Unspecified
	• Open Message Error
	  • Unsupported Version
	  • Bad Peer As
	  • Bad BGP Identifier
	  • Unsupported Optional Parameter
	  • Authentication Failure
	  • Unacceptable Hold Time
	  • Unspecified
	• Update Message Error
	  • Malformed Attribute List
	  • Unrecognized Attribute
	  • Missing Attribute
	  • Attribute Flag Error
	  • Attribute Length Error
	  • Invalid Origin Attribute
	  • Invalid NextHop Attribute
	  • Optional Attribute Error
	  • Invalid Network Field
	  • Malformed AS Path
	  • Unspecified
	• Hold Timer Expired
	• Finite State Machine Error
	• Cease
	• Unspecified

Notification Received	See above.
Neighbor NLRI Negotiation	The state of the 's NLRI negotiation with the neighbor. The states can include the following: • Peer negotiated IPv6 unicast capability. • Peer configured for IPv6 unicast routes. • Peer negotiated IPv4 unicast capability. • Peer negotiated IPv4 multicast capability.

TABLE 355 BGP4+ neighbor configuration information and statistics (Continued)

This field...	Displays...
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IPv6 address of the .
Local port	The TCP port the is using for the BGP4+ TCP session with the neighbor.
Remote host	The IPv6 address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4+ TCP session with the .
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the that have not been acknowledged by the neighbor.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.

TABLE 355 BGP4+ neighbor configuration information and statistics (Continued)

This field...	Displays...
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying routes advertised to a BGP4+ neighbor

You can display a summary or detailed information about the following:

- All routes a has advertised to a neighbor.
- A specified route a has advertised to a neighbor.

For example, to display a summary of all routes a has advertised to neighbor 2000:4::110, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2000:4::110 advertised-routes
       There are 2 routes advertised to neighbor 2000:4::110
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
       Prefix              Next Hop        Metric     LocPrf     Weight Status
1      2002:1234::/32      ::              1                     32768  BL
         AS_PATH:
2      2002::/16           ::              1                     32768  BL
         AS_PATH:
```

Syntax: **show ipv6 bgp neighbor** <*ipv6-address*> **advertised-routes [detail]** <*ipv6-prefix*>/<*prefix-length*>

The <*ipv6-address*> parameter displays routes advertised to a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **detail** keyword displays detailed information about the advertised routes. If you do not specify this keyword, a summary of the advertised routes displays.

The <*ipv6-prefix*>/<*prefix-length*> parameter displays the specified route advertised to the neighbor only. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

This display shows the following information.

TABLE 356 Summary of route information advertised to a BGP4+ neighbor

This field...	Displays...
Number of BGP4+ Routes advertised to specified neighbor (appears only in display for all routes)	The number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The advertised route's prefix.
Next Hop	The next-hop for reaching the advertised route from the .
Metric	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.

TABLE 356 Summary of route information advertised to a BGP4+ neighbor (Continued)

This field...	Displays...
LocPrf	The degree of preference for the advertised route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference range is 0 – 4294967295.
Weight	The value that this associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Status	The advertised route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4+ has determined that this is the optimal route to the destination. • b – NOT-INSTALLED-BEST – BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • E – EBGP. The route was learned through a in another AS. • I – IBGP. The route was learned through a in the same AS. • L – LOCAL. The route originated on this .
AS-PATH	The AS-path information for the route.

For example, to display details about all routes a has advertised to neighbor 2000:4::110, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2000:4::110 advertised-routes detail
There are 2 routes advertised to neighbor 2000:4::110
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
1       Prefix: 2002:1234::/32,  Status: BL,  Age: 6d13h28m7s
         NEXT_HOP: 2000:4::106,  Learned from Peer: Local Router
          LOCAL_PREF: none,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:
            Adj_RIB_out count: 1,  Admin distance 190
2       Prefix: 2002::/16,  Status: BL,  Age: 6d13h31m22s
         NEXT_HOP: 2000:4::106,  Learned from Peer: Local Router
          LOCAL_PREF: none,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:

            Adj_RIB_out count: 1,  Admin distance 190
```

This display shows the following information.

TABLE 357 Detailed route information advertised to a BGP4+ neighbor

This field...	Displays...
Number of BGP4+ Routes advertised to specified neighbor (appears only in display for all routes)	For information about this field, refer to Table 356 on page 1933.
Status codes	For information about this field, refer to Table 356 on page 1933.
Prefix	For information about this field, refer to Table 356 on page 1933.

TABLE 357 Detailed route information advertised to a BGP4+ neighbor (Continued)

This field...	Displays...
Status	For information about this field, refer to Table 356 on page 1933.
Age	The age of the advertised route, in seconds.
Next Hop	For information about this field, refer to Table 356 on page 1933.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. “Local Router” indicates that the itself learned the route.
LOCAL_PREF	For information about this field, refer to Table 356 on page 1933.
MED	The value of the advertised route’s MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP4+ through EGP. • IGP – The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	For information about this field, refer to Table 356 on page 1933.
AS-PATH	The AS-path information for the route.
Adj RIB out count	The number of routes in the ’s current BGP4+ Routing Information Base (Adj-RIB-Out) for a specified neighbor.
Admin distance	The administrative distance of the route.

Displaying BGP4+ neighbor route-attribute entries

The route-attribute entries table lists sets of BGP4+ attributes stored in the ’s memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the typically has fewer route attribute entries than routes.

For example, to display the route-attribute entries table for a BGP4+ neighbor 2000:4::110, enter the following command.

```
NetIron# show ipv6 bgp neighbor 2000:4::110 attribute-entries
               Total number of BGP Attribute Entries: 1
1       Next Hop  :2000:4::106        Metric   :1              Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0         Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :65001
        Address: 0x26579354  Hash:332 (0x0301fcd4) Reference Counts: 2:0:0
```

Syntax: show ipv6 bgp neighbor <*ipv6-address*> **attribute-entries**

The <*ipv6-address*> parameter displays the route attribute entries for a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

This display shows the following information.

TABLE 358 BGP4+ neighbor route-attribute entries information

This field...	Displays...
Total number of BGP Attribute Entries	The number of route attribute entries for the specified neighbor.
Next Hop	The IPv6 address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP4+ through EGP. • IGP – The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • `AS Number` shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • `Router-ID` shows the that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated *and* this aggregation has resulted in information loss: • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred • None – Indicates the attribute is not present. Note: Information loss under these circumstances is a normal part of BGP4+ and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	For debugging purposes only.
Hash	For debugging purposes only.
Reference Counts	For debugging purposes only.

Displaying route flap dampening statistics for a BGP4+ neighbor

To display route flap dampening statistics for a specified BGP4+ neighbor, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2000:4::110 flap-statistics
Total number of flapping routes: 14
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flaps Since    Reuse    Path
h>  2001:2::/32        166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
*>  3892:34::/32       166.90.213.77   1     0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ipv6 bgp neighbor <*ipv6-address*> **flap-statistics**

The <*ipv6-address*> parameter displays the route flap dampening statistics for a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

This display shows the following information.

TABLE 359 Route flap dampening statistics for a BGP4+ neighbor

This field...	Displays...
Total number of flapping routes	The total number of routes in the neighbor's BGP4+ route table that have changed state and thus have been marked as flapping routes.
Status code	Indicates the status of the route, which can be one of the following: • > – This is the best route among those in the neighbor's BGP4+ route table to the route's destination. • d – This route is currently dampened, and thus unusable. • h – The route has a history of flapping and is unreachable now. • * – The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The IPv6 address of the advertising peer.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time (in hh:mm:ss) since the first flap of this route.
Reuse	The amount of time (in hh:mm:ss) after which the path is again available.
Path	The AS path of the route.

You also can display all the dampened routes by using the **show ipv6 bgp dampened-paths** command. For more information, refer to "Displaying dampened BGP4+ paths" on page 1920.

Displaying last error packet from a BGP4+ neighbor

You can display information about the last packet that contained an error from any of a 's neighbors. The displayed information includes the error packet's contents decoded in a human-readable format.

For example, to display information about the last error packet from any of a 's neighbors, enter the following command.

```
NetIron# show ipv6 bgp neighbor last-packet-with-error
    Total number of BGP Neighbors: 266
No received packet with error logged for any neighbor
```

Syntax: show ipv6 bgp neighbor last-packet-with-error

This display shows the following information.

TABLE 360 Last error packet information for BGP4+ neighbors

This field...	Displays...
Total number of BGP Neighbors	The total number of configured neighbors for a .
Last error	The error packet's contents decoded in a human-readable format or notification that no packets with an error were received.

Displaying Outbound Route Filters received from a BGP4+ neighbor

You can display the Outbound Route Filters (ORFs) received from a BGP4+ neighbor. This option applies to cooperative route filtering feature.

For example, to display the ORFs received from neighbor 2000:2::110, enter the following command.

```
NetIron# show ipv6 bgp neighbor 2000:2::110 received prefix-filter
ip prefix-list 2000:2::110: 4 entries
     seq 5 permit 3000:3::45/16 ge 18 le 28
     seq 10 permit 4000:4::88/24
     seq 15 permit 5000:5::37/8 le 32
     seq 20 permit 6000:6::83/16 ge 18
```

Syntax: show ipv6 bgp neighbor <*ipv6-address*> **received prefix-filter**

The <*ipv6-address*> parameter displays the prefix filter learned from a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Displaying routes received from a BGP4+ neighbor

You can display a summary or detailed route information received in route updates from a specified BGP4+ neighbor since you enabled the soft reconfiguration feature.

For example, to display a summary of the route information received in route updates from neighbor 2000:4::10, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2:2:2:2:: received-routes
There are 4 received routes from neighbor 2:2:2:2::
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
    PrefixNext HopMetricLocPrfWeightStatus
1   2002::/642:2:2:2:: 01000BE
AS_PATH: 400
2   2003::/642:2:2:2:: 11000BE
AS_PATH: 400
3   2004::/642:2:2:2:: 11000BE
AS_PATH: 400
4   2005::/642:2:2:2:: 11000BE
AS_PATH: 400
```

Syntax: show ipv6 bgp neighbor <*ipv6-address*> **received-routes [detail]**

The <*ipv6-address*> parameter displays route information received from a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **detail** keyword displays detailed route information. If you do not specify this parameter, a summary of route information displays.

This display shows the following information.

TABLE 361 Summary of route information received from a BGP4+ neighbor

This field...	Displays...
Number of BGP4+ Routes received from a neighbor	The number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The received route's prefix.
Next Hop	The IPv6 address of the next that is used when forwarding a packet to the received route.
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the advertised route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
Weight	The value that this associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Status	The advertised route's status, which can be one or more of the following: A – AGGREGATE. The route is an aggregate route for multiple networks. B – BEST. BGP4+ has determined that this is the optimal route to the destination. b – NOT-INSTALLED-BEST – BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. E – EBGP. The route was learned through a in another AS. H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. I – IBGP. The route was learned through a in the same AS. L – LOCAL. The route originated on this . M – MULTIPATH. BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. F – FILTERED. This route was filtered out by BGP4+ route policies on the , but the saved updates containing the filtered routes.

For example, to display details about routes received from neighbor 2000:1:1::1, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2000:1:1::1 received-routes detail
There are 4 received routes from neighbor 2000:1:1::1
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
1 Prefix: 1000:1:1::/64, Status: BI, Age: 0h17m25s
NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
AS_PATH:
Adj_RIB_out count: 1, Admin distance 200
2 Prefix: 2000:1:1::/64, Status: I, Age: 0h17m25s
NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
AS_PATH:
3 Prefix: 2000:1:11::1/128, Status: BI, Age: 0h17m25s
NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
LOCAL_PREF: 100, MED: 0, ORIGIN: igp, Weight: 0
AS_PATH:
Adj_RIB_out count: 1, Admin distance 200
4 Prefix: 2000:1:17::/64, Status: BI, Age: 0h17m25s
NEXT_HOP: 2000:1:1::1, Learned from Peer: 2000:1:1::1 (100)
LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
AS_PATH:
Adj_RIB_out count: 1, Admin distance 200
```

This display shows the following information.

TABLE 362 Detailed route information received from a BGP4+ neighbor

This field...	Displays...
Number of BGP4+ routes received from a neighbor	For information about this field, refer to Table 361 on page 1939.
Status codes	For information about this field, refer to Table 361 on page 1939.
Prefix	For information about this field, refer to Table 361 on page 1939.
Status	For information about this field, refer to Table 361 on page 1939.
Age	The age of the route, in seconds.
Next hop	The next-hop router for reaching the route from the .
Learned from peer	The IPv6 address of the neighbor from which this route is learned. “Local Router” indicates that the itself learned the route.
Local pref	For information about this field, refer to Table 361 on page 1939.
MED	The value of the route’s MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP4+ through EGP. • IGP – The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	For information about this field, refer to Table 361 on page 1939.
AS Path	For information about this field, refer to Table 361 on page 1939.

TABLE 362 Detailed route information received from a BGP4+ neighbor (Continued)

This field...	Displays...
Adj RIB out count	The number of routes in the 's current BGP4+ Routing Information Base (Adj-RIB-Out) for a specified neighbor.
Admin distance	The administrative distance of the route.

Displaying the Adj-RIB-Out for a BGP4+ neighbor

You can display a summary or detailed information about the following:

- All routes in a 's current BGP4+ Routing Information Base (Adj-RIB-Out) for a specified neighbor.
- A specified route in a 's current BGP4+ RIB for a specified neighbor.

The RIB contains the routes that the either has most recently sent to the neighbor or is about to send to the neighbor.

For example, to display a summary of all routes in a 's RIB for neighbor 2000:4::110, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2000:4::110 rib-out-routes
       There are 2 RIB_out routes for neighbor 2000:4::110
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
       Prefix              Next Hop        Metric     LocPrf     Weight Status
1      2002:1234::/32      ::              1          100        32768  BL
         AS_PATH:
2      2002::/16           ::              1          100        32768  BL
         AS_PATH:
```

Syntax: **show ipv6 bgp neighbor** <*ipv6-address*> **rib-out-routes** [<*ipv6-prefix*>/<*prefix-length*> | **detail** [<*ipv6-prefix*>/<*prefix-length*> <*network-mask*>]]

The <*ipv6-address*> parameter displays the RIB routes for a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*ipv6-prefix*>/<*prefix-length*> parameter displays the specified RIB route for the neighbor. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter.

The **detail** <*ipv6-prefix*>/<*prefix-length*> <*network-mask*> parameter displays detailed information about the specified RIB routes. If you do not specify this parameter, a summary of the RIB routes displays. You must specify the <*ipv6-prefix*> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <*prefix-length*> parameter as a decimal value. A slash mark (/) must follow the <*ipv6-prefix*> parameter and precede the <*prefix-length*> parameter. You must specify the <*network-mask*> parameter using 8-bit values in dotted decimal notation.

This display shows the following information.

TABLE 363 Summary of RIB route information for a BGP4+ neighbor

This field...	Displays...
Number of RIB_out routes for a specified neighbor (appears only in display for all RIB routes)	The number of RIB routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The RIB route's prefix.
Next Hop	The next-hop router for reaching the route from the .
Metric	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
Weight	The value that this associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Status	The RIB route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4+ has determined that this is the optimal route to the destination. E – EBGP. The route was learned through a in another AS. • I – IBGP. The route was learned through a in the same AS. • L – LOCAL. The route originated on this .
AS-PATH	The AS-path information for the route.

For example, to display details about all RIB routes for neighbor 2000:4::110, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2000:4::110 rib-out-routes detail
                    There are 2 RIB_out routes for neighbor 2000:4::110
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
1       Prefix: 2002:1234::/32,  Status: BL,  Age: 6d18h17m53s
         NEXT_HOP: ::,  Learned from Peer: Local Router
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:
            Adj_RIB_out count: 1,  Admin distance 190
2       Prefix: 2002::/16,  Status: BL,  Age: 6d18h21m8s
         NEXT_HOP: ::,  Learned from Peer: Local Router
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:

            Adj_RIB_out count: 1,  Admin distance 190
            Adj_RIB_out count: 1,  Admin distance 190
```

This display shows the following information.

TABLE 364 Detailed RIB route information for a BGP4+ neighbor

This field...	Displays...
Number of RIB_out routes for a specified neighbor (appears only in display for all routes)	For information about this field, refer to Table 363 on page 1942.
Status codes	For information about this field, refer to Table 363 on page 1942.
Prefix	For information about this field, refer to Table 363 on page 1942.
Status	For information about this field, refer to Table 363 on page 1942.
Age	The age of the RIB route, in seconds.
Next Hop	For information about this field, refer to Table 363 on page 1942.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. “Local Router” indicates that the itself learned the route.
LOCAL_PREF	For information about this field, refer to Table 363 on page 1942.
MED	The value of the RIB route’s MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP4+ through EGP. • IGP – The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	For information about this field, refer to Table 363 on page 1942.
AS-PATH	For information about this field, refer to Table 363 on page 1942.

Displaying the best and unreachable routes received from a BGP4+ neighbor

You can display a summary or detailed information about the following types of BGP4+ routes received from a specified neighbor:

- **Best routes** – The “best” routes to their destinations, which are installed in the ’s IPv6 route table.
- **Unreachable** – The routes whose destinations are unreachable using any of the BGP4+ paths in the IPv6 route table.

For example, to display a summary of the best routes to a destination received from neighbor 2000:4::106, enter the following command.

```
NetIron# show ipv6 bgp neighbor 2000:4::106 routes best
       There are 2 accepted routes from neighbor 2000:4::106
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      2002::/16          2000:4::106     1          100        0      BE
         AS_PATH: 65001
2      2002:1234::/32     2000:4::106     1          100        0      BE
         AS_PATH: 65001
```

Syntax: show ipv6 bgp neighbor <*ipv6-address*> **routes best | detail [best | unreachable] | unreachable**

The <*ipv6-address*> parameter displays the routes for a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **best** keyword displays the "best" routes, which are installed in the IPv6 route table.

The **unreachable** keyword displays the routes whose destinations are unreachable using any of the BGP4+ paths in the IPv6 route table.

The **detail** keyword displays detailed information about the routes. If you do not specify this parameter, a summary of the routes displays.

This display shows the following information.

TABLE 365 Summary of best and unreachable routes from a BGP4+ neighbor

This field...	Displays...
Number of accepted routes from a specified neighbor	The number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The route's prefix.
Next Hop	The next-hop router for reaching the route from the .
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
Weight	The value that this associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.

TABLE 365 Summary of best and unreachable routes from a BGP4+ neighbor (Continued)

This field...	Displays...
Status	The route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4+ has determined that this is the optimal route to the destination. • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • E – EBGP. The route was learned through a in another AS. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – IBGP. The route was learned through a in the same AS. • L – LOCAL. The route originated on this . • M – MULTIPATH. BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE:** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. • F – FILTERED. This route was filtered out by BGP4+ route policies on the , but the saved updates containing the filtered routes.
AS-PATH	The AS-path information for the route.

For example, to display detailed information about the best routes to a destination received from neighbor 2000:4::106, enter the following command.

```
NetIron# show ipv6 bgp neighbor 2000:4::106 routes detail best
       There are 2 accepted routes from neighbor 2000:4::106
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
1       Prefix: 2002::/16,  Status: BE,  Age: 18h48m56s
         NEXT_HOP: 2000:4::106,  Learned from Peer: 2000:4::106 (65001)
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 0
           AS_PATH: 65001
2       Prefix: 2002:1234::/32,  Status: BE,  Age: 18h48m56s
         NEXT_HOP: 2000:4::106,  Learned from Peer: 2000:4::106 (65001)
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 0
           AS_PATH: 65001
```

This display shows the following information.

TABLE 366 Detailed best and unreachable routes from a BGP4+ neighbor

This field...	Displays...
Number of accepted routes from a specified neighbor (appears only in display for all routes)	For information about this field, refer to Table 365 on page 1944.
Status codes	For information about this field, refer to Table 365 on page 1944.
Prefix	For information about this field, refer to Table 365 on page 1944.
Status	For information about this field, refer to Table 365 on page 1944.

TABLE 366 Detailed best and unreachable routes from a BGP4+ neighbor (Continued)

This field...	Displays...
Age	The age of the route, in seconds.
Next Hop	For information about this field, refer to Table 365 on page 1944.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the itself learned the route.
LOCAL_PREF	For information about this field, refer to Table 365 on page 1944.
MED	The value of the RIB route's MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP4+ through EGP. • IGP – The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	For information about this field, refer to Table 365 on page 1944.
AS-PATH	For information about this field, refer to Table 365 on page 1944.

Displaying IPv6 neighbor route summary information

You can display route summary information for all neighbors or a specified neighbor only.

For example, to display summary information for neighbor 2000:4::110, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp neighbor 2000:4::110 routes-summary
1   IP Address: 2000:4::110
Routes Accepted/Installed:0,  Filtered/Kept:0,  Filtered:0
   Routes Selected as BEST Routes:0
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0

NLRIs Received in Update Message:0,  Withdraws:0 (0),  Replacements:0
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0

Routes Advertised:2,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:2,  Withdraws:0,  Replacements:0

Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0 Outbound Routes Holder:0
```

Syntax: show ipv6 bgp neighbor [<*ipv6-address*>] **routes-summary**

This display shows the following information.

TABLE 367 BGP4+ neighbor route summary information

This field...	Displays...
IP Address	The IPv6 address of the neighbor
Routes Received	How many routes the has received from the neighbor during the current BGP4+ session: • Accepted or Installed – Indicates how many of the received routes the accepted and installed in the BGP4+ route table. • Filtered or Kept – Indicates how many routes were filtered out, but were nonetheless retained in memory for use by the soft reconfiguration feature. • Filtered – Indicates how many of the received routes were filtered out.
Routes Selected as BEST Routes	The number of routes that the selected as the best routes to their destinations.
BEST Routes not Installed in IPv6 Forwarding Table	The number of routes received from the neighbor that are the best BGP4+ routes to their destinations, but were nonetheless not installed in the IPv6 route table because the received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the does not have a valid RIPng, OSPFv3, or static IPv6 route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages: • Withdraws – The number of withdrawn routes the has received. • Replacements – The number of replacement routes the has received.
NLRIs Discarded due to	Indicates the number of times the discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The 's configured maximum prefix amount had been reached. • AS Loop – An AS loop occurred. An AS loop occurs when the BGP4+ AS-path attribute contains the local AS number. • Invalid Nexthop Address – The next hop value was not acceptable. • Duplicated Originator_ID – The originator ID was the same as the local router ID. • Cluster_ID – The cluster list contained the local cluster ID, or contained the local router ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the has advertised to this neighbor: • To be Sent – The number of routes the has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the has queued up to send to this neighbor in UPDATE messages.

TABLE 367 BGP4+ neighbor route summary information (Continued)

This field...	Displays...
NLRIs Sent in Update Message	The number of NLRIs for new routes the has sent to this neighbor in UPDATE messages: • Withdraws – The number of routes the has sent to the neighbor to withdraw. • Replacements – The number of routes the has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the has run out of BGP4+ memory for the neighbor during the current BGP4+ session: • Receiving Update Messages – The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes(NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes – The number of times there was no memory for BGP4+ attribute entries. • Outbound Routes (RIB-out) – The number of times there was no memory to place a “best” route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised. • Outbound Routes Holder – For debugging purposes only.

Displaying BGP4+ peer group configuration information

You can display configuration information for all peer groups or a specified peer group configured on a .

For example, to display configuration information for a peer group named peer1, enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp peer-group peer1
1   BGP peer-group is pg1, Remote AS: 65002
    Description: device group 1
       NextHopSelf: yes
      Address family : IPV4 Unicast
      Address family : IPV4 Multicast
      Address family : IPV6 Unicast
    Members:
       IP Address: 192.169.102.2
       IP Address: 192.169.100.2
       IP Address: 192.169.101.2
       IP Address: 192.169.103.2
       IP Address: 192.169.104.2
       IP Address: 192.169.105.2
       IP Address: 192.169.106.2
       IP Address: 192.169.107.2
       IP Address: 192.169.108.2
       IP Address: 192.169.109.2
       IP Address: 192.169.110.2
       IP Address: 192.169.111.2
       IP Address: 192.169.112.2
```

Syntax: show ipv6 bgp peer-group [<*peer-group-name*>]

The display shows only parameters that have values different from their default settings.

Displaying BGP4+ summary

To view summary BGP4+ information for the , enter the following command at any level of the CLI.

```
NetIron# show ipv6 bgp summary
BGP4 Summary
Router ID: 223.223.223.223   Local AS Number : 65001
Confederation Identifier : not configured
Confederation Peers:
Maximum Number of Paths Supported for Load Sharing : 1
Number of Neighbors Configured : 1
Number of Routes Installed : 2
Number of Routes Advertising to All Neighbors : 2
Number of Attribute Entries Installed : 1
Neighbor Address  AS#   State   Time     Rt:Accepted Filtered Sent   ToSend
2000:4::110       65002 ESTAB  21h32m32s    0        0        2      0
```

Syntax: show ipv6 bgp summary

This display shows the following information.

TABLE 368 BGP4+ summary information

This field...	Displays...
Router ID	The 's router ID.
Local AS Number	The BGP4+ AS number in which the resides.
Confederation Identifier	The AS number of the confederation in which the resides.
Confederation Peers	The numbers of the local ASs contained in the confederation. This list matches the confederation peer list you configure on the .
Maximum Number of Paths Supported for Load Sharing	The maximum number of route paths across which the can balance traffic to the same destination. The feature is enabled by default but the default number of paths is 1. You can increase the number from 2 – 8 paths.
Number of Neighbors Configured	The number of BGP4+ neighbors configured on this .
Number of Routes Installed	The number of BGP4+ routes in the 's BGP4+ route table. To display the BGP4+ route table, refer to "Displaying the BGP4+ route table" on page 1911.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors.
Number of Attribute Entries Installed	The number of BGP4+ route-attribute entries in the 's route-attributes table. To display the route-attribute table, refer to "Displaying BGP4+ route-attribute entries" on page 1918.
Neighbor Address	The IPv6 addresses of this 's BGP4+ neighbors.
AS#	The AS number.

TABLE 368 BGP4+ summary information (Continued)

This field...	Displays...

State	The state of this 's neighbor session with each neighbor. The states are from this 's perspective of the session, not the neighbor's perspective. The state values can be one of the following for each :
	• IDLE – The BGP4+ process is waiting to be started. Usually, enabling BGP4+ or establishing a neighbor session starts the BGP4+ process.
	  • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• ADMND – The neighbor has been administratively shut down.
	  • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• CONNECT – BGP4+ is waiting for the connection process for the TCP neighbor session to be completed.
	• ACTIVE – BGP4+ is waiting for a TCP connection from the neighbor.
	NOTE: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.
	• OPEN SENT – BGP4+ is waiting for an Open message from the neighbor.
	• OPEN CONFIRM – BGP4+ has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle.
	• ESTABLISHED – BGP4+ is ready to exchange UPDATE packets with the neighbor.
	  • If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed.
	NOTE: If you display information for the neighbor using the **show ipv6 bgp neighbor** <*ipv6-address*> command, the TCP receiver queue value will be greater than 0.

Time	The time that has passed since the state last changed.
Accepted	The number of routes received from the neighbor that this installed in the BGP4+ route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this filtered out some of the routes received in the UPDATE messages.
Filtered	The routes or prefixes that have been filtered out. • If soft reconfiguration is enabled, this field shows how many routes were filtered out (not placed in the BGP4+ route table) but retained in memory. • If soft reconfiguration is not enabled, this field shows the number of BGP4+ routes that have been filtered out.
Sent	The number of BGP4+ routes that the has sent to the neighbor.
ToSend	The number of routes the has queued to send to this neighbor.

Chapter

Configuring IPv6 Multicast Features

46

PowerConnect B-MLXe supports the IPv6 Multicast features:

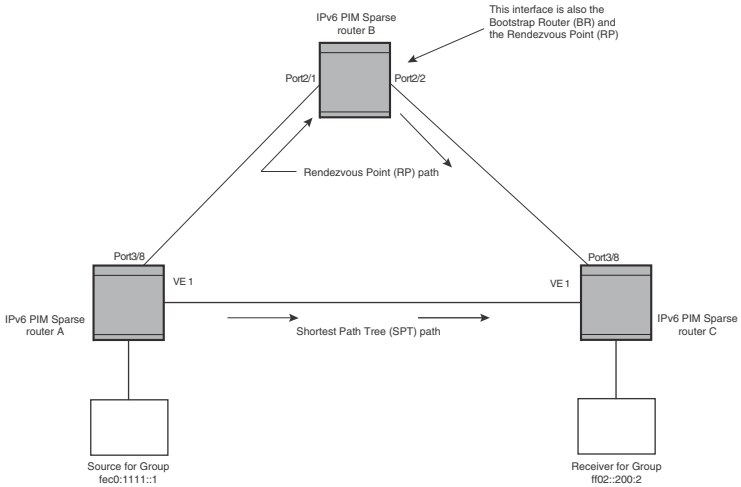
- IPv6 PMRI
- IPv6 PIM-SSM
- IPv6 PIM Sparse
- IPv6 PIM Anycast RP
- Multicast Listener Discovery (MLD)
- PIM and MLD Snooping for IPv6
- VRF Support for IPv6 Multicast

IPv6 PIM Sparse

IPv6 Protocol Independent Multicast (PIM) Sparse is supported. IPv6 PIM Sparse provides multicasting that is especially suitable for widely distributed multicast environments.

In an IPv6 PIM Sparse network, an IPv6 PIM Sparse router that is connected to a host that wants to receive information for a multicast group must explicitly send a join request on behalf of the receiver (host).

FIGURE 220 Example IPv6 PIM Sparse domain



PIM Sparse router types

Routers that are configured with PIM Sparse interfaces also can be configured to fill one or more of the following roles:

- BSR – The Bootstrap Router (BSR) distributes RP information to the other PIM Sparse routers within the domain. Each PIM Sparse domain has one active BSR. For redundancy, you can configure ports on multiple routers as candidate BSRs. The PIM Sparse protocol uses an election process to select one of the candidate BSRs as the BSR for the domain. The BSR with the highest BSR priority (a user-configurable parameter) is elected. If the priorities result in a tie, then the candidate BSR interface with the highest IP address is elected. In the example in Figure 220, PIM Sparse router B is the BSR. Port 2/2 is configured as a candidate BSR.
- RP – The Rendezvous Points (RP) is the meeting point for PIM Sparse sources and receivers. A PIM Sparse domain can have multiple RPs, but each PIM Sparse multicast group address can have only one active RP. PIM Sparse routers learn the addresses of RPs and the groups for which they are responsible from messages that the BSR sends to each of the PIM Sparse routers. In the example in Figure 220, PIM Sparse router B is the RP. Port 2/2 is configured as a candidate Rendezvous Point (RP).

  To enhance overall network performance, the device uses the RP to forward only the first packet from a group source to the group receivers. After the first packet, the device calculates the shortest path between the receiver and the source (the Shortest Path Tree, or SPT) and uses the SPT for subsequent packets from the source to the receiver. The device calculates a separate SPT for each source-receiver pair.

NOTE
It is recommended that you configure the same ports as candidate BSRs and RPs.

RP paths and SPT paths

Figure 220 shows two paths for packets from the source for group fec0:1111::1 and a receiver for the group. The source is attached to PIM Sparse router A and the recipient is attached to PIM Sparse router C. PIM Sparse router B is the RP for this multicast group. As a result, the default path for packets from the source to the receiver is through the RP. However, the path through the RP sometimes is not the shortest path. In this case, the shortest path between the source and the receiver is over the direct link between router A and router C, which bypasses the RP (router B).

To optimize PIM traffic, the protocol contains a mechanism for calculating the Shortest Path Tree (SPT) between a given source and a receiver. PIM Sparse routers can use the SPT as an alternative to using the RP for forwarding traffic from a source to a receiver. By default, the device forwards the first packet it receives from a given source to a given receiver using the RP path, but subsequent packets from that source to that receiver through the SPT. In Figure 220, router A forwards the first packet from group fec0:1111::1 source to the destination by sending the packet to router B, which is the RP. Router B then sends the packet to router C. For the second and all future packets that router A receives from the source for the receiver, router A forwards them directly to router C using the SPT path.

RFC 3513 and RFC 4007 compliance for IPv6 multicast scope-based forwarding

The IPv6 multicast implementation recognizes scopes and conforms to the scope definitions in RFC 3513. Per RFC 3513, scopes 0 and 3 are reserved and packets are not forwarded with an IPv6 destination multicast address of scopes 0 and 3. Additionally, scopes 1 and 2 are defined as Node-Local and Link-Local and are not forwarded. Thus, the implementation forwards only those packets with an IPv6 multicast destination address with scope 4 or higher.

RFC 4007 defines ‘scope zones’ and requires that the forwarding of packets received on any interface of a particular scope zone be restricted to that scope zone. Currently, the device supports one zone for each scope, and the default zone for scope 4 and higher consists of all interfaces in the system. Thus, the default zones for scope 4 and higher are the same size.

Configuring PIM Sparse

To configure the device for IPv6 PIM Sparse, perform the following tasks:

- Enable the IPv6 PIM Sparse of multicast routing.
- Configure an IPv6 address on the interface.
- Enable IPv6 PIM Sparse.
- Identify the interface as an IPv6 PIM Sparse border, if applicable.
- Enable IPv6 Protocol Independent Multicast Sparse mode (PIM-SM) for a specified VRF, if applicable.
- Identify the device as a candidate PIM Sparse Bootstrap Router (BSR), if applicable.
- Identify the device as a candidate PIM Sparse Rendezvous Point (RP), if applicable.
- Specify the IP address of the RP (if you want to statically select the RP).

NOTE
It is recommended that you configure the same device as both the BSR and the RP.

IPv6 PIM-Sparse mode

To configure a device for IPv6 PIM Sparse, perform the following tasks:

- Identify the Layer 3 switch as a candidate sparse Rendezvous Point (RP), if applicable.
- Specify the IPv6 address of the RP (to configure statically).

The following example enables IPv6 PIM-SM routing. Enter the following command at the configuration level to enable IPv6 PIM-SM globally.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)#
```

To enable IPv6 PIM Sparse mode on an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 2/2
NetIron(config-if-e10000-2/2)# ipv6 address a000:1111::1/64
NetIron(config-if-e10000-2/2)# ipv6 pim-sparse
```

Syntax: [no] ipv6 pim-sparse

Use the **no** option to remove IPv6 PIM sparse configuration from the interface.

The commands in this example add an IPv6 interface to port 2/2, then enable IPv6 PIM Sparse on the interface.

Configuring IPv6 PIM-SM on a virtual routing interface

You can enable IPv6 PIM-SM on a virtual routing interface by entering commands such as the following.

```
NetIron(config)# interface ve 15
NetIron(config-vif-15)# ipv6 address a000:1111::1/64
NetIron(config-vif-15)# ipv6 pim-sparse
```

Enabling IPv6 PIM-SM for a specified VRF

To enable IPv6 PIM-SM for the VRF named “blue”, create the VRF named “blue”, enable it for IPv6 routing, and then enable IPv6 PIM-SM for the VRF, as shown in the following example.

```
NetIron(config)# vrf blue
NetIron(config-vrf-blue)# rd 11:1
NetIron(config-vrf-blue)# address-family ipv6
NetIron(config-vrf-blue-ipv6)# router pim
NetIron(config-pim-router)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

The **vrf** parameter allows you to configure IPv6 PIM-SM on the virtual routing instance (VRF) specified by the <*vrf-name*> variable. All PIM parameters available for the default router instance are configurable for a VRF-based PIM instance.

Use the **no** option to remove all configuration for PIM multicast on the specified VRF.

Configuring BSRs

In addition to the global and interface parameters configured in the prior sections, you must identify an interface on at least one device as a candidate PIM Sparse Bootstrap Router (BSR) and a candidate PIM Sparse Rendezvous Point (RP).

NOTE
It is possible to configure the device as only a candidate BSR or an RP, but it is recommended that you configure the same interface on the same device as both a BSR and an RP.

To configure the device as a candidate BSR, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ip6-pim-router)# bsr-candidate ethernet 1/3 32 64
BSR address: 31::207, hash mask length: 32, priority: 64
```

This command configures Ethernet interface 1/3 as the BSR candidate with a mask length of 32 and a priority of 64.

To configure the device as a candidate BSR for a specified VRF, enter the commands as shown in the following example.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# bsr-candidate ethernet 1/3 32 64
BSR address: 31::207, hash mask length: 32, priority: 64
```

Syntax: [**no**] **bsr-candidate ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> <*hash-mask-length*> [<*priority*>]

Use the **no** option to remove the candidate BSR configuration for a specified VRF.

The **ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> parameter specifies the interface. The device will advertise the specified interface's IP address as a candidate BSR:

- Enter **ethernet** <*slot*>/<*portnum*> for a physical interface (port).
- Enter **loopback** <*num*> for a loopback interface.
- Enter **ve** <*num*> for a virtual interface.

The <*hash-mask-length*> variable specifies the number of bits in a group address that are significant when calculating the group-to-RP mapping. You can specify a value from 1 through 32.

The <*priority*> variable specifies the BSR priority. You can specify a value from 0 through 255. When the election process for BSR takes place, the candidate BSR with the highest priority becomes the BSR. The default is 0.

Setting the BSR message interval

The BSR message interval timer defines the interval at which the BSR sends RP candidate data to all IPv6-enabled routers within the IPv6 PIM Sparse domain. The default is 60 seconds.

To set the IPv6 PIM BSR message interval timer to 16 seconds, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# bsr-msg-interval 16
Changed BSR message interval to 16 seconds.
```

To set the IPv6 PIM BSR message interval timer to 16 seconds for a specified VRF, enter the commands as shown in the following example.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# bsr-msg-interval 16
Changed BSR message interval to 16 seconds.
```

Syntax: [**no**] **bsr-msg-interval** <*num*>

The <*num*> parameter specifies the number of seconds and can be from 10 – 65535. The default is 60.

Use the **no** option to disable a timer that has been configured.

Configuring candidate RP

Enter a command such as the following to configure the device as a candidate RP.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# rp-candidate ethernet 2/2
```

To configure the device as a candidate RP for a specified VRF, enter the commands as shown in the following example.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# rp-candidate ethernet 2/2
```

Syntax: [**no**] **rp-candidate ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> | **pos** <*slot*>/<*portnum*>

The **ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> l **pos** <*slot*>/<*portnum*> parameter specifies the interface. The device will advertise the specified interface IP address as a candidate RP:

- Enter **ethernet** <*slot*>/<*portnum*> for a physical interface (port).
- Enter **loopback** <*num*> for a loopback interface.
- Enter **ve** <*num*> for a virtual interface.
- Enter **pos** <*slot/portnum*> specifies the individual POS interface described by the <slot/port> variable.

To add address ranges for which the device is a candidate RP, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# rp-candidate add ff02::200:2 64
```

To add address ranges for a specified VRF for which the device is a candidate RP, enter commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# rp-candidate add ff02::200:2 64
```

Syntax: [**no**] **rp-candidate add** <*group-ipv6 address*> <*mask-bits*>

You can delete the configured RP candidate group ranges by entering commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# rp-candidate delete ff02::200:1 128
```

You can delete the configured RP candidate group ranges for a specified VRF by entering commands such as the following:

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router-vrf-blue)# rp-candidate delete ff02::200:1 128
```

Syntax: [**no**] **rp-candidate delete** <*group-ipv6 address*> <*mask-bits*>

The usage for the <*group-ipv6 address*> <*mask-bits*> parameter is the same as for the **rp-candidate add** command.

Statically specifying the RP

It is recommended that you use the IPv6 PIM Sparse mode RP election process so that a backup RP can automatically take over if the active RP router becomes unavailable. However, if you do not want the RP to be selected by the RP election process but instead you want to explicitly identify the RP by its IPv6 address, use the **rp-address** command.

If you explicitly specify the RP, the device uses the specified RP for all group-to-RP mappings and overrides the set of candidate RPs supplied by the BSR.

NOTE
Specify the same IP address as the RP on all IPv6 PIM Sparse routers within the IPv6 PIM Sparse domain. Make sure the device is on the backbone or is otherwise well-connected to the rest of the network.

To specify the IPv6 address of the RP, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# rp-address 31::207
```

The command in the previous example identifies the router interface at IPv6 address 31:207 as the RP for the IPv6 PIM Sparse domain. The device will use the specified RP and ignore group-to-RP mappings received from the BSR.

To specify the IPv6 address of the RP for a specified VRF, enter commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# rp-address 31::207
```

Syntax: [**no**] **rp-address** <*ipv6-addr*>

The <*ipv6-addr*> parameter specifies the IPv6 address of the RP.

Updating IPv6 PIM Sparse forwarding entries with a new RP configuration

If you make changes to your static RP configuration, the entries in the IPv6 PIM Sparse multicast forwarding table continue to use the old RP configuration until they are aged out.

The **clear IPv6 pim rp-map** command allows you to update the entries in the static multicast forwarding table immediately after making RP configuration changes. This command is meant to be used with the **rp-address** command.

To update the entries in an IPv6 PIM Sparse static multicast forwarding table with a new RP configuration, enter the following command at the privileged EXEC level of the CLI.

```
NetIron(config)# clear ipv6 pim rp-map
```

Syntax: clear ipv6 pim rp-map

Embedded Rendezvous Point

Global deployment of IPv4 multicast relies on Multicast Source Discovery Protocol (MSDP) to convey information about the active sources. Because IPv6 provides more address space, the RP address can be included in the multicast group address.

NOTE
The IPv6 group address must be part of the FF70:/12 prefix.

Embedded RP support is enabled by default. You can disable it using the following commands.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# no rp-embedded
```

To disable embedded RP support for a specified VRF, enter the following commands.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# no rp-embedded
```

Syntax: [**no**] **rp-embedded**

Changing the Shortest Path Tree threshold

In a typical IPv6 PIM Sparse domain, there may be two or more paths from a designated router (DR) for a multicast source to an IPv6 PIM group receiver:

- **Path through the RP** – This is the path the device uses the first time it receives traffic for an IPv6 PIM group. However, the path through the RP may not be the shortest path from the device to the receiver.

- **Shortest Path** – Each IPv6 PIM Sparse router that is a DR for an IPv6 receiver calculates a short path tree (SPT) towards the source of the IPv6 multicast traffic. The first time the device configured as an IPv6 PIM router receives a packet for an IPv6 group, it sends the packet to the RP for that group, which in turn will forward it to all the intended DRs that have registered with the RP. The first time the device is a recipient, it receives a packet for an IPv6 group and evaluates the shortest path to the source and initiates a switchover to the SPT. Once the device starts receiving data on the SPT, the device proceeds to prune itself from the **RPT**.

By default, the device switches from the RP to the SPT after receiving the first packet for a given IPv6 PIM Sparse group. The device maintains a separate counter for each IPv6 PIM Sparse source-group pair.

You can change the number of packets the device receives using the RP before switching to using the SPT.

To change the number of packets the device receives using the RP before switching to the SPT, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# spt-threshold 1000
```

To change the number of packets the device receives using the RP before switching to the SPT for a specified VRF, enter commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# spt-threshold 1000
```

Syntax: [**no**] **spt-threshold infinity** | <*num*>

The **infinity** | <*num*> parameter specifies the number of packets. If you specify **infinity**, the device sends packets using the RP indefinitely and does not switch over to the SPT. If you enter a specific number of packets, the device does not switch over to using the SPT until it has sent the number of packets you specify using the RP.

Setting the RP advertisement interval

To specify how frequently the candidate RP configured on the device sends candidate RP advertisement messages to the BSR, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# rp-adv-interval 180
Changed RP ADV interval to 180 seconds.
```

To specify how frequently the candidate RP configured on the device sends candidate RP advertisement messages to the BSR for a specified VRF, enter commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# rp-adv-interval 180
Changed RP ADV interval to 180 seconds.
```

Syntax: **rp-adv-interval** <*seconds*>

The <*seconds* > parameter specifies the number of seconds. The default is 60 seconds.

Route selection precedence for multicast

The **route-precedence** command allows the user to specify a precedence table that dictates how routes are selected for multicast.

NOTE
PIM must be enabled at the global level.

Configuring the route precedence by specifying the route types

The **route-precedence** {mc-non-default mc-default uc-non-default uc-default none} command allows you to control the selection of routes based on the route types. There are four different types of routes:

- Non-default route from the mRTM
- Default route from the mRTM
- Non-default route from the uRTM
- Default route from the uRTM

Using the **route-precedence** command, you may specify an option for all of the precedence levels.

To specify a non-default route from the mRTM, then a non-default route from the uRTM, then a default route from the mRTM, and then a default route from the uRTM, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# route-precedence mc-non-default uc-non-default
mc-default uc-default
```

The **none** option can be used to fill up the precedence table in order to ignore certain types of routes. To use the unicast default route for multicast, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# route-precedence mc-non-default mc-default
uc-non-default none
```

To specify a non-default route from the mRTM, then a non-default route from the uRTM, then a default route from the mRTM, and then a default route from the uRTM for a specified VRF, enter commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# route-precedence mc-non-default
uc-non-default mc-default uc-default
```

The **none** option can be used to fill up the precedence table in order to ignore certain types of routes. To use the unicast default route for multicast for a specified VRF, enter commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# route-precedence mc-non-default
mc-default uc-non-default none
```

Syntax: [**no**] **route-precedence** {**mc-non-default** | **mc-default** | **uc-non-default** | **uc-default** | **none**}

The default value is the **route-precedence mc-non-default mc-default uc-non-default uc-default** command.

Use the **mc-non-default** parameter to specify a multicast non-default route.

Use the **mc-default** parameter to specify a multicast default route.

Use the **uc-non-default p**arameter to specify a unicast non-default route.

Use the **uc-default** parameter to specify a unicast default route.

Use the **none** parameter to ignore certain types of routes.

The **no** form of this command removes the configuration.

Changing the PIM Join and Prune message interval

By default, the device sends PIM Sparse Join or Prune messages every 60 seconds. These messages inform other PIM Sparse routers about clients who want to become receivers (Join) or stop being receivers (Prune) for PIM Sparse groups.

NOTE
Use the same Join or Prune message interval on all the PIM Sparse routers in the PIM Sparse domain. If the routers do not all use the same timer interval, the performance of PIM Sparse can be adversely affected.

To change the Join or Prune interval, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# message-interval 30
```

To change the Join or Prune interval for a specified VRF, enter the commands as shown in the following example.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# message-interval 30
```

Syntax: [**no**] **message-interval** <*seconds*>

The <*seconds*> parameter specifies the number of seconds and can be from 1 through 65535 seconds. The default is 60 seconds.

Modifying neighbor timeout

Neighbor timeout is the interval after which a PIM router will consider a neighbor to be absent. If the timer expires before receiving a new hello message, the PIM router will time out the neighbor.

To apply an IPv6 PIM neighbor timeout value of 33 seconds to all ports on the router operating with PIM, enter the commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# nbr-timeout 33
```

To apply an IPv6 PIM neighbor timeout value of 33 seconds for a specified VRF operating with PIM, enter the commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# nbr-timeout 33
```

Syntax: [**no**] **nbr-timeout** <*seconds*>

The <*seconds*> parameter specifies the number of seconds. The valid range is from 35 through 65535 seconds.

Setting the prune wait interval

The **prune-wait** command allows you to set the amount of time the PIM router should wait for a join override before pruning an Outgoing Interface List Optimization (OIF) from the entry.

To change the default join override time to 2 seconds, enter commands such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# prune-wait 2
```

To change the default join override time to 2 seconds for a specified VRF, enter commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# prune-wait 2
```

Syntax: [**no**] **prune-wait** <*seconds*>

The <*seconds*> parameter specifies the number of seconds. The valid range is from 0 through 30 seconds. The default is 3 seconds.

Setting the register suppress interval

The **register-suppress-time** command allows you to set the amount of time the PIM router uses to periodically trigger the NULL register message.

NOTE
The register suppress time configuration applies only to the first hop PIM router.

To change the default register suppress time to 90 seconds, enter commands such as the following:

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# register-suppress-time 90
```

To change the default register suppress time to 90 seconds for a specified VRF, enter commands such as the following:

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# register-suppress-time 90
```

Syntax: [**no**] **register-suppress-time** <*seconds*>

The <*seconds*> parameter specifies the number of seconds. The valid range is from 60 through 120 seconds. The default is 60 seconds.

Setting the register probe time

The **register-probe-time** command allows you to set the amount of time the PIM router waits for a register-stop from an RP before it generates another NULL register to the PIM RP. The register probe time configuration applies only to the first hop PIM router.

NOTE
Once a PIM first hop router successfully registers with a PIM RP, the PIM first hop router will not default back to the data registration. All subsequent registers will be in the form of the NULL registration.

To change the default register probe time to 20 seconds, enter commands such as following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# register-probe-time 20
```

To change the default register probe time to 20 seconds for a specified VRF, enter commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# register-probe-time 20
```

Syntax: [**no**] **register-probe-time** <*seconds*>

The <*seconds*> parameter specifies the number of seconds. The valid range is from 10 through 50 seconds. The default is 10 seconds.

Setting the inactivity timer

The router deletes a forwarding entry if the entry is not used to send multicast packets. The IPv6 PIM inactivity timer defines how long a forwarding entry can remain unused before the router deletes it.

To apply an IPv6 PIM inactivity timer of 160 seconds to all IPv6 PIM interfaces, enter the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# inactivity-timer 160
```

To apply an IPv6 PIM inactivity timer of 160 seconds for a specified VRF, enter the commands as shown in the following example.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# inactivity-timer 160
```

Syntax: [**no**] **inactivity-timer** <*seconds*>

The <*seconds*> parameter specifies the number of seconds. The valid range is 60 through 3600 seconds. The default is 180 seconds.

Changing the hello timer

The hello timer defines the interval at which periodic hellos are sent out to PIM interfaces. Routers use hello messages to inform neighboring routers of their presence. To change the hello timer, enter a command such as the following.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# hello-timer 62
```

To change the hello timer for a specified VRF, enter the commands as shown in the following example.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# hello-timer 62
```

Syntax: [**no**] **hello-timer** <*seconds*>

The <*seconds*> parameter specifies the number of seconds. The valid range is 10 through 3600 seconds. The default is 60 seconds.

Enabling Source-specific Multicast

Using the Any-Source Multicast (ASM) service model, sources and receivers register with a multicast address. The protocol uses regular messages to maintain a correctly configured broadcast network where all sources can send data to all receivers and all receivers get broadcasts from all sources.

With Source-specific Multicast (SSM), the “channel” concept is introduced where a “channel” consists of a single source and multiple receivers that specifically register to get broadcasts from that source. Consequently, receivers are not burdened with receiving data they have no interest in, and network bandwidth requirements are reduced because the broadcast need only go to a subset of users. The address range ff30:/12 has been assigned by the Internet Assigned Numbers Authority (IANA) for use with SSM.

SSM simplifies IPv6 PIM-SM by eliminating the RP and all protocols related to the RP.

Configuring Source-specific Multicast

IPv6 PIM-SM must be enabled on any ports on which you want SSM to operate. Enter the **ssm-enable** command under the IPv6 router PIM level to globally enable SSM filtering.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# ssm-enable ff02::200:2
```

To enable SSM for a specified VRF, enter the commands as shown in the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ssm-enable ff02::200:2
```

Syntax: [**no**] **ssm-enable** [**range** <*address-range*>]

The **range** <*address-range*> option allows you to define the SSM range of IPv6 multicast addresses.

Modifying the Hop-Limit threshold

The Time To Live (TTL) defines the minimum value required in a packet in order for the packet to be forwarded out the interface. For example, if the Hop-Limit for an interface is set at 10, it means that only those packets that ingress with a TTL value of 11 or more will be forwarded out the TTL-10 interface. Thus, with a default Hop-Limit threshold of 1, only packets ingressing with a Hop-Limit of 2 or greater will be forwarded out. Note that the Hop-Limit threshold only applies to routed interfaces. Switched interfaces ignore the Hop-Limit threshold. Possible Hop-Limit values are from 1 through 64. The default Hop-Limit value is 1. To configure a Hop-Limit of 45, enter the following command.

```
NetIron(config-if-e10000-3/24)# ipv6 pim ttl-threshold 45
```

To configure a Hop-Limit of 45 on a virtual Ethernet interface, enter the following commands.

```
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ipv6 pim ttl-threshold 45
```

Syntax: ipv6 pim ttl-threshold <1-64>

Configuring a DR priority

The DR priority option lets a network administrator give preference to a particular router in the DR election process by giving it a numerically higher DR priority. To set a DR priority higher than the default value of 1, use the **ipv6 pim dr-priority** command as shown in the example below.

```
NetIron(config-if-e10000-3/24)# ipv6 pim dr-priority 50
```

To set a DR priority higher than the default value of 1 on a virtual Ethernet interface, use the **ipv6 pim dr-priority** command as shown in the following.

```
NetIron(config)# interface ve 10
```

```
NetIron(config-vif-10)# ipv6 pim dr-priority 50
```

Syntax: [**no**] **ipv6 pim dr-priority** <*priority-value*>

The <*priority-value*> variable is the value that you want to set for the DR priority. The range of values is from 0 through 65535. The default value is 1.

The **no** option removes the command and sets the DR priority back to the default value of 1.

The following information may be useful for troubleshooting:

- If more than one router has the same DR priority on a subnet (as in the case of default DR priority on all), the router with the numerically highest IP address on that subnet will get elected as the DR.
- The DR priority information is used in the DR election *only if all* the PIM routers connected to the subnet support the DR priority option. If there is at least one PIM router on the subnet that does not support this option, then the DR election falls back to the backwards compatibility mode in which the router with the numerically highest IP address on the subnet is declared the DR regardless of the DR priority values.

Passive Multicast Route Insertion

To prevent unwanted multicast traffic from being sent to the CPU, IPv6 PIM routing and Passive Multicast Route Insertion (PMRI) can be used together to ensure that multicast streams are only forwarded out ports with interested receivers and unwanted traffic is dropped in hardware on Layer 3 routers.

PMRI enables a Layer 3 switch running IPv6 PIM Sparse to create an entry for a multicast route (for example, (S,G)), with no directly attached clients or when connected to another PIM router (transit network).

When a multicast stream has no output interfaces, the Layer 3 switch can drop packets in hardware if the multicast traffic meets the following conditions in IPv6 PIM-SM.

- The route has no OIF.
- The directly connected source passes source RPF check and completes data registration with the RP, or the non-directly connected source passes source RPF check.

If the OIF is inserted after the hardware-drop entries are installed, the hardware entries will be updated to include the OIFs.

NOTE
Disabling hardware-drop does not immediately take away existing hardware-drop entries, they will go through the normal route aging processing when the traffic stops.

Configuring PMRI

PMRI is enabled by default. To disable PMRI, enter the following commands.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# hardware-drop-disable
```

To disable PMRI for a specified VRF, enter the commands as shown in the following example.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# hardware-drop-disable
```

Syntax: [**no**] **hardware-drop-disable**

Displaying hardware-drop

Use the **show ipv6 pim sparse** command to display if the hardware-drop feature has been enabled or disabled.

```
NetIron# show ipv6 pim sparse
Global PIM Sparse Mode Settings
  Hello interval        : 30          Neighbor timeout                    : 105
  Bootstrap Msg interval: 60          Candidate-RP Advertisement interval: 60
  Join/Prune interval   : 60          SPT Threshold                       : 1
  SSM Enabled: Yes
  SSM Group Range: ff30::/12
  Hardware Drop Enabled : Yes
```

Displaying PIM Sparse configuration information and statistics

You can display the following PIM Sparse information:

- Basic PIM Sparse configuration information
- IPv6 interface information
- Group information
- BSR information
- Candidate RP information
- RP-to-group mappings
- RP information for an IPv6 PIM Sparse group
- RP set list
- Multicast neighbor information
- The IPv6 PIM multicast cache
- IPv6 PIM RPF
- IPv6 PIM counters
- IPv6 PIM resources
- IPv6 PIM traffic statistics

Displaying basic PIM Sparse configuration information

To display IPv6 PIM Sparse configuration information, enter the **show ipv6 pim sparse** command at any CLI level.

```
NetIron# show ipv6 pim sparse
Global PIM Sparse Mode Settings
  Hello interval            : 30       Neighbor timeout                    : 105
  Bootstrap Msg interval    : 60       Candidate-RP Advertisement interval: 60
  Register Suppress interval: 60       Register Stop Delay                 : 60
  Join/Prune interval       : 60       SPT Threshold                       : 1
  Inactivity interval       : 180      Hardware Drop Enabled               : Yes
  SSM Enabled               : Yes
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **sparse**

The **vrf** parameter allows you to configure IPv6 PIM on the virtual routing instance (VRF) specified by the <*vrf-name*> variable.

Table 369 displays the output from the **show ipv6 pim sparse** command.

TABLE 369 Output from the **show ipv6 pim sparse** command

Field	Description
Global PIM Sparse mode settings	
Hello interval	How frequently the device sends IPv6 PIM Sparse hello messages to its IPv6 PIM Sparse neighbors. This field shows the number of seconds between hello messages. IPv6 PIM Sparse routers use hello messages to discover one another.
Neighbor timeout	How many seconds the device will wait for a hello message from a neighbor before determining that the neighbor is no longer present and removing cached IPv6 PIM Sparse forwarding entries for the neighbor.
Bootstrap Msg interval	How frequently the BSR configured on the device sends the RP set to the RPs within the IPv6 PIM Sparse domain. The RP set is a list of candidate RPs and their group prefixes. The group prefix of a candidate RP indicates the range of IPv6 PIM Sparse group numbers for which it can be an RP. **NOTE:** This field contains a value only if an interface on the device is elected to be the BSR. Otherwise, the field is blank.
Candidate-RP Advertisement interval	How frequently the candidate PR configured on the device sends candidate RP advertisement messages to the BSR. **NOTE:** This field contains a value only if an interface on the device is configured as a candidate RP. Otherwise, the field is blank.
Join or Prune interval	How frequently the device sends IPv6 PIM Sparse Join or Prune messages for the multicast groups it is forwarding. This field shows the number of seconds between Join or Prune messages. The device sends Join or Prune messages on behalf of multicast receivers that want to join or leave an IPv6 PIM Sparse group. When forwarding packets from IPv6 PIM Sparse sources, the device sends the packets only on the interfaces on which it has received join requests in Join or Prune messages for the source group.
SPT Threshold	The number of packets the device sends using the path through the RP before switching to using the SPT path.
Inactivity Interval	How long a forwarding entry can remain unused before the router deletes it.
SSM Enabled	If yes, source-specific multicast is configured globally on this router.
IPv6 PIM Sparse interface information **NOTE:** You also can display IPv6 multicast interface information using the **show ipv6 pim interface** command.	
Interface	The type of interface and the interface number. The interface type can be one of the following: • Ethernet • VE The number is either a port number (and slot number if applicable) or the virtual interface (VE) number.

TABLE 369 Output from the **show ipv6 pim sparse** command (Continued)

Field	Description
TTL Threshold	Following the TTL threshold value, the interface state is listed. The interface state can be one of the following: • Disabled • Enabled
Local Address	Indicates the IP address configured on the port or virtual interface.

Displaying IPv6 PIM interface information

You can display IPv6 PIM multicast interface information using the **show ipv6 pim interface** command.

```
NetIron# show ipv6 pim
Interface v30
  PIM Version : V2 MODE : PIM SM
  TTL Threshold: 1, Enabled
  DR: fe80::20c:dbff:fef6:a00 on e3/2
  Link Local Address: fe80::20c:dbff:fef5:e900
  Global Address: 1e1e::4

Interface v167
  PIM Version : V2 MODE : PIM SM
  TTL Threshold: 1, Enabled
  DR: itself
  Link Local Address: fe80::20c:dbff:fef5:e900
  Global Address: a7a7::1
Interface l1
  PIM Version : V2 MODE : PIM SM
  TTL Threshold: 1, Enabled
  DR: itself
  Link Local Address: fe80::20c:dbff:fef5:e900
  Global Address: 8c8c::4
```

To display IPv6 PIM multicast interface information for a specified VRF, enter the following command as shown in the example.

```
NetIron# show ipv6 pim vrf2 interface

---------+-------------------------------------------------+----+---+---+---------+-------+------
Interface|Global Address                                    |Ver |St |TTL|Multicast|  VRF  | DR
         |  + Designated Router                       Port |    |   |Thr|Boundary |       | Prio
---------+-------------------------------------------------+----+---+---+---------+-------+------
     e3/6 101::1                                           SMv2 Ena   1 None      vrf2    1
          + Itself
    e3/16 102::1                                           SMv2 Ena   1 None      vrf2    1
          + Itself
       l1 100::1                                           SMv2 Ena   1 None      vrf2    1
          + Itself
```

Syntax: **show ipv6 pim** [**vrf** <*vrf-name*>] **interface** [**ethernet** <*slot/port*> | **loopback** <*number*> |**pos** <*slot/port*> | **ve** <*number*>]

The **vrf** parameter allows you to display IPv6 multicast interface information for the VRF instance identified by the <*vrf-name*> variable.

The **ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> parameter specifies the IPv6 PIM multicast interface.

- Enter **ethernet** <*slot*>/<*portnum*> for a physical interface (port).
- Enter **loopback** <*num*> for a loopback interface.
- Enter **ve** <*num*> for a virtual interface.
- Enter **pos** <*slot/portnum*> specifies the individual POS interface described by the <slot/port> variable.

Displaying a list of multicast groups

To display IPv6 PIM group information, enter the **show ipv6 pim group** command at any CLI level.

```
NetIron# show ipv6 pim group
Total number of groups: 1
1    Group ff7e:a40:2001:3e8:27:0:1:2  Ports
     Group member at  e3/1: v31
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **group**

The **vrf** parameter allows you to display IPv6 PIM group information for the VRF instance identified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 pim group** command.

TABLE 370 Output from the **show ipv6 pim group** command

Field	Description
Total number of Groups	Lists the total number of IPv6 multicast groups the device is forwarding.
Group	The multicast group address.
Ports	The device ports connected to the receivers of the groups.

Displaying BSR information

To display information on a device that has been elected as the BSR, enter the **show ipv6 pim bsr** command at the CLI level.

```
NetIron# show ipv6 pim bsr
PIMv2 Bootstrap information

This system is the elected Bootstrap Router (BSR)
  BSR address: 2001:3e8:255:255::17
  Uptime: 00:12:09, BSR priority: 0, Hash mask length: 126
  Next bootstrap message in 00:00:30

Next Candidate-RP-advertisment in 00:00:30
  RP: 2001:3e8:255:255::17
    group prefixes:
    ff00:: / 8

  Candidate-RP-advertisement period: 60
```

The following example shows information displayed on a device that is not the BSR. Notice that some fields shown in the previous example do not appear in the following example.

```
NetIron# show ipv6 pim bsr
PIMv2 Bootstrap information
 BSR address  = 2001:3e8:255:255::17
 BSR priority = 0
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **bsr**

The **vrf** parameter allows you to display IPv6 PIM BSR information for the VRF instance identified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 pim bsr** command.

TABLE 371 Output from the **show ipv6 pim bsr** command

Field	Description
BSR address	The IPv6 address of the interface configured as the IPv6 PIM Sparse Bootstrap Router (BSR).
Uptime	The amount of time the BSR has been running. **NOTE:** This field appears only if this device is the BSR.
BSR priority	The priority assigned to the interface for use during the BSR election process. During BSR election, the priorities of the candidate BSRs are compared and the interface with the highest BSR priority becomes the BSR.
Hash mask length	The number of significant bits in the IPv6 multicast group comparison mask. This mask determines the IPv6 multicast group numbers for which the device can be a BSR. The default is 32 bits, which allows the device to be a BSR for any valid IPv6 multicast group number. **NOTE:** This field appears only if this device is a candidate BSR.
Next bootstrap message in	Indicates how many seconds will pass before the BSR sends its next Bootstrap message. **NOTE:** This field appears only if this device is the BSR.
Next Candidate-RP-advertisement message in	Indicates how many seconds will pass before the BSR sends its next candidate RP advertisement message. **NOTE:** This field appears only if this device is a candidate BSR.
RP	Indicates the IPv6 address of the Rendezvous Point (RP). **NOTE:** This field appears only if this device is a candidate BSR.
group prefixes	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP. **NOTE:** This field appears only if this device is a candidate BSR.
Candidate-RP-advertisement period	Indicates how frequently the BSR sends candidate RP advertisement messages. **NOTE:** This field appears only if this device is a candidate BSR.

Displaying candidate RP information

To display candidate RP information, enter the **show ipv6 rp-candidate** command at any CLI level.

```
NetIron# show ipv6 pim rp-candidate

Next Candidate-RP-advertisement in 00:00:10
  RP: 1be::11:21
    group prefixes:
    ff00:: / 8

  Candidate-RP-advertisement period: 60
```

This example shows information displayed on a device that is a candidate RP. The following example shows the message displayed on a device that is not a candidate RP.

```
NetIron# show ipv6 pim rp-candidate

This system is not a Candidate-RP.
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **rp-candidate**

The **vrf** parameter allows you to display IPv6 candidate RP information for the VRF instance identified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 pim rp-candidate** command.

TABLE 372 Output from the **show ipv6 pim rp-candidate** command

Field	Description
Candidate-RP-advertisement in	Indicates how many seconds will pass before the BSR sends its next RP message. **NOTE:** This field appears only if this device is a candidate RP.
RP	Indicates the IPv6 address of the Rendezvous Point (RP). **NOTE:** This field appears only if this device is a candidate RP.
group prefixes	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP. **NOTE:** This field appears only if this device is a candidate RP.
Candidate-RP-advertisement period	Indicates how frequently the BSR sends candidate RP advertisement messages. **NOTE:** This field appears only if this device is a candidate RP.

Displaying RP-to-group mappings

To display RP-to-group-mappings, enter the **show ipv6 pim rp-map** command at any CLI level.

```
NetIron# show ipv6 pim rp-map
Idx Group address                          RP address

  1ff1e::1:2 2001:3e8:255:255::17
  2              ff7e:a40:2001:3e8:27:0:1:2 2001:3e8:27::a
  3              ff7e:140:2001:3e8:16:0:1:2 2001:3e8:16::1
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **rp-map**

The **vrf** parameter allows you to display IPv6 RP-to-group-mappings for the VRF instance identified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 rp-map** command.

TABLE 373 Output from the show ipv6 pim rp-map command

Field	Description
Index	The index number of the table entry in the display.
Group address	Indicates the IPv6 PIM Sparse multicast group address using the listed RP.
RP address	Indicates the Iv6 address of the Rendezvous Point (RP) for the listed PIM Sparse group.

Displaying RP information for an IPv6 PIM Sparse group

To display RP information for an IPv6 PIM Sparse group, enter the following command at any CLI level.

```
NetIron# show ipv6 pim rp-hash ff1e::1:2
  RP: 2001:3e8:255:255::17, v2
    Info source: 2001:3e8:255:255::17, via bootstrap
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **rp-hash** <*group-addr*>

The **vrf** parameter allows you to display RP information for a PIM Sparse group for the VRF instance identified by the <*vrf-name*> variable.

The <*group-addr*> parameter is the address of an IPv6 PIM Sparse IP multicast group.

Table 374 displays the output from the **show ipv6 pim rp-hash** <*group-addr*> command.

Output from the show ipv6 pin rp-hash <group-addr> command

TABLE 374 Output from the **show ipv6 pin rp-hash** <*group-addr*> command

Field	Description
RP	Indicates the IPv6 address of the Rendezvous Point (RP) for the specified IPv6 PIM Sparse group. Following the IPv6 address is the port or virtual interface through which this device learned the identity of the RP.
Info source	Indicates the IPv6 address on which the RP information was received. Following the IPv6 address is the method through which this device learned the identity of the RP.

Displaying the RP set list

To display the RP set list, enter the **show ipv6 pim rp-set** command at any CLI level.

```
NetIron# show ipv6 pim rp-set
Static RP
---------
Static RP count: 1
100::1
Number of group prefixes Learnt from BSR: 0
No RP-Set present
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **rp-set**

The **vrf** parameter allows you to display the RP set for the VRF instance identified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 pim rp-set** command.

TABLE 375 Output from the **show ipv6 pim rp-set** command

Field	Description
Number of group prefixes	The number of IPv6 PIM Sparse group prefixes for which the RP is responsible.
Group prefix	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP.
RPs expected or received	Indicates how many RPs were expected and received in the latest Bootstrap message.
RP <*num*>	Indicates the RP number. If there are multiple RPs in the IPv6 PIM Sparse domain, a line of information for each of them is listed, and they are numbered in ascending numerical order.
priority	The RP priority of the candidate RP. During the election process, the candidate RP with the highest priority is elected as the RP.
age	The age (in seconds) of this RP-set. **NOTE:** If this device is not a BSR, this field contains zero. Only the BSR ages the RP-set.

Displaying multicast neighbor information

To display information about IPv6 PIM neighbors, enter the **show ipv6 pim neighbor** command at any CLI level.

```
NetIron# show ipv6 pim neighbor
Port Phy_Port Neighbor                              Holdtime Age   UpTime Priority
                                                    sec      sec   sec
e11/15 e11/15 fe80::45:27:49:4                      105       20    1010   1
v312   e11/3  fe80::45:27:1:2                       105       10    1900   40
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **neighbor**

The **vrf** parameter allows you to display the IPv6 PIM neighbors for the VRF instance identified by the <*vrf-name*> variable.

Table 376 displays the output from the **show ipv6 pim neighbor** command.

TABLE 376 Output from the **show ipv6 pim neighbor** command

Field	Description
Port	The interface through which the device is connected to the neighbor.
Neighbor	The IPv6 interface of the IPv6 PIM neighbor interface.
Holdtime sec	Indicates how many seconds the neighbor wants this device to hold the entry for this neighbor in memory. The neighbor sends the Hold Time in its hello packets. • If the device receives a new hello packet before the Hold Time received in the previous packet expires, the device updates its table entry for the neighbor. • If the device does not receive a new hello packet from the neighbor before the Hold time expires, the device assumes the neighbor is no longer available and removes the entry for the neighbor.
Age sec	The number of seconds since the device received the last hello message from the neighbor.

TABLE 376 Output from the **show ipv6 pim neighbor** command (Continued)

Field	Description
UpTime sec	The number of seconds the PIM neighbor has been up. This timer starts when the device receives the first hello messages from the neighbor.
Priority	The DR priority that is used in the DR election process. This can be a configured value or the default value of 1.

Displaying the IPv6 PIM multicast cache

To display the IPv6 PIM multicast cache, enter the **show ipv6 pim mcache** command at any CLI level.

```
NetIron# show ipv6 pim mcache
Total 4 entries
Free mll entries: 766
1    (*, ff7e:140:2001:3e8:16:0:1:2) RP2001:3e8:16::1 in NIL, cnt=0
     Sparse Mode, RPT=1 SPT=0 Reg=0
     No upstream neighbor because RP 2001:3e8:16::1 is itself
     num_oifs = 1 v312
     L3 (SW) 1: e3/15(VL312)
     Flags fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 needRte=0
     age=0 fid: 0405, mvid      1
2    (2001:3e8:0:170::101, ff7e:140:2001:3e8:16:0:1:2) in v23 (e3/23), cnt=2
     Sparse Mode, RPT=0 SPT=1 Reg=0
     upstream neighbor=fe80::45:0:160:4
     num_oifs = 0
     Flags fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 needRte=0
     age=0 fid: 0402, mvid     23
3    (2001:3e8:0:170::101, ff7e:a40:2001:3e8:27:0:1:2) in v23 (e3/23), cnt=0
     Sparse Mode, RPT=0 SPT=1 Reg=0
     upstream neighbor=fe80::45:0:160:4
     num_oifs = 1 v31
     L3 (HW) 1: e3/1(VL31)
     Flags fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 needRte=0
     age=0 fid: 0404, mvid     23
4    (*, ff7e:a40:2001:3e8:27:0:1:2) RP2001:3e8:27::a in v312, cnt=0
     Sparse Mode, RPT=1 SPT=0 Reg=0
     upstream neighbor=fe80::45:27:1:3
     num_oifs = 1 v31
     L3 (SW) 1: e3/1(VL31)
     Flags fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 needRte=0
     age=0 fid: 0406, mvid    312

Total number of mcache entries 4
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **mcache**

The **vrf** parameter allows you to display the IPv6 PIM multicast cache for the VRF instance identified by the <*vrf-name*> variable.

Displaying IPv6 PIM RPF

The **show ipv6 pim rpf** command displays what PIM sees as the reverse path to the source. While there may be multiple routes back to the source, the one displayed by the **show ipv6 pim rpf** command is the one that PIM thinks is best.

```
NetIron# show ipv6 pim vrf eng rpf 130.50.11.10
Source 130.50.11.10 directly connected on e4/1
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **rpf** <*ip-address*>

The **vrf** parameter allows you to display what PIM sees as the reverse path to the source for a VRF instance specified by the <*vrf-name*> variable.

The <*ip-address*> variable specifies the source address for RPF check.

Displaying IPv6 PIM counters

You can display the number of default-vlan-id changes that have occurred since the applicable VRF was created and how many times a tagged port was placed in a VLAN since the applicable VRF was created, as shown in the following example.

```
NetIron(config)# show ipv6 pim vrf eng counter
Event Callback:
   DFTVlanChange   :          0               VlanPort       :           2

LP to MP IPCs:
   SM_REGISTER    :           0               MCAST_CREATE     :          31
   S_G_AGEOUT      :          4               WRONG_IF       :           0
   ABOVE_THRESHOLD :          0               MCAST_FIRST_DATA :          31
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **counter**

The **vrf** parameter allows you to display IPv6 PIM counters for the VRF instance identified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 vrf eng counter** command.

TABLE 377 Output from the **show ipv6 pim vrf eng counter** command

Field	Description
DFTVlanChange	The number of default-vlan-id changes that have occurred since the applicable VRF was created.
VlanPort	The number of times that a tagged port was placed in a VLAN since the applicable VRF was created.

Displaying the IPv6 PIM resources

To display the hardware resource information, such as hardware allocation, availability, and limit for software data structure, enter the **show ipv6 pim resource** command.

```
NetIron# show ipv6 pim resource
                        allocated    in-use available allo-fail  up-limit
NBR list                   64         1        63         0       512
Static RP                  64         0        64         0        64
Anycast RP                 64         0        64         0        64
timer                      64         0        64         0  no-limit
prune                      32         0        32         0  no-limit
pimsm J/P elem          12240         0     12240         0     48960
pimsm OIF                  64        60         4         0  no-limit
mcache                     64        60         4         0  no-limit
mcache hash link          997        60       937         0  no-limit
graft if no mcache        197         0       197         0  no-limit
groups                     64         0        64         0  no-limit
group-memberships          64         0        64         0  no-limit
sources                   256         0       256         0  no-limit
client sources            256         0       256         0  no-limit
pim/dvm intf. group        64         0        64         0  no-limit
pim/dvm global group       64         0        64         0  no-limit
MLD Resources:
  groups                   64         0        64         0  no-limit
  phy-ports                64         0        64         0  no-limit
  exist-phy-port          256         0       256         0  no-limit
group-query              256         0       256         0  no-limit
  group-query              256         0       256         0  no-limit
Hardware-related Resources:
HW MVID: 0 allocated for MCAST6 of total allocated 1
Total (S,G) entries 30
Total SW FWD entries 0
  Total sw w/Tag MVID entries 0
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **resource**

The **vrf** parameter allows you to display IPv6 hardware resource information for the VRF instance identified by the <*vrf-name*> variable.

Table 378 displays the output from the **show ipv6 pim resource** command.

TABLE 378 Output from the **show ipv6 pim resource** command

Field	Description
alloc	Number of nodes of that data that are currently allocated in memory.
in-use	Number of allocated nodes in use.
avail	Number of allocated nodes are not in use.
allo-fail	Number of allocated notes that failed.
up-limit	Maximum number of nodes that can be allocated for a data structure. This may or may not be configurable, depending on the data structure

Displaying PIM traffic statistics

To display IPv6 PIM traffic statistics, enter the **show ipv6 pim traffic** command at any CLI level.

```
NetIron# show ipv6 pim traffic
Port    Hello           Join          Prune          Assert
     [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]
MLD Statistics:
  Total Recv/Xmit 356/161
  Total Discard/chksum  0/0
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] **traffic**

The **vrf** parameter allows you to display IPv6 traffic statistics for the VRF instance identified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 pim traffic** command.

TABLE 379 Output from the **show ipv6 pim traffic** command

Field	Description
Port	The port or virtual interface on which the IPv6 PIM interface is configured.
Hello	The number of IPv6 PIM Hello messages sent or received on the interface.
J or P	The number of Join or Prune messages sent or received on the interface. **NOTE:** Unlike PIM dense, PIM Sparse uses the same messages for Joins and Prunes.
Register	The number of Register messages sent or received on the interface.
RegStop	The number of Register Stop messages sent or received on the interface.
Assert	The number of Assert messages sent or received on the interface.
Total Recv or Xmit	The total number of IGMP messages sent and received by the device.
Total Discard or chksum	The total number of IGMP messages discarded, including a separate counter for those that failed the checksum comparison.

Clearing the IPv6 PIM forwarding cache

You can clear the IPv6 PIM forwarding cache using the **clear ipv6 pim cache** command.

```
NetIron# clear ipv6 pim cache
```

Syntax: clear ipv6 pim [**vrf** <*vrf-name*>] **cache**

Use the **vrf** parameter to clear the IPv6 PIM forwarding cache for a VRF instance specified by the <*vrf-name*> variable.

Clearing the IPv6 PIM message counters

You can clear the IPv6 PIM message counters using the **clear ipv6 pim counters** command.

```
NetIron# clear ipv6 pim counters
```

Syntax: clear ipv6 pim [**vrf** <*vrf-name*>] **counters**

Use the **vrf** parameter to clear the IPv6 PIM message counters for a VRF instance specified by the <*vrf-name*> variable.

Updating PIM Sparse forwarding entries with a new RP configuration

If you make changes to your static RP configuration, the entries in the IPv6 PIM Sparse multicast forwarding table continue to use the old RP configuration until they are aged out.

The **clear IPv6 pim rp-map** command allows you to update the entries in the static multicast forwarding table immediately after making RP configuration changes. This command is meant to be used with **rp-address** command.

To update the entries in an IPv6 PIM Sparse static multicast forwarding table with a new RP configuration, enter the **clear ipv6 pim rp-map** command at the privileged EXEC level of the CLI.

```
NetIron(config)# clear ipv6 pim rp-map
```

Syntax: clear ipv6 pim [**vrf** <*vrf-name*>] **rp-map**

Use the **vrf** parameter to clear the IPv6 PIM Sparse static multicast forwarding table for a VRF instance specified by the <*vrf-name*> variable.

Clearing the IPv6 PIM traffic

To clear counters on IPv6 PIM traffic, enter the **clear ipv6 pim traffic** command.

```
NetIron# clear ipv6 pim traffic
```

Syntax: clear ipv6 pim [**vrf** <*vrf-name*>] **traffic**

Use the **vrf** par meter to clear counters on IPv6 PIM traffic for a VRF instance specified by the <*vrf-name*> variable.

Setting the maximum number of IPv6 multicast routes supported

You can use the **ipv6 max-mroute** command to define the maximum number of IPv6 multicast routes supported. The default VRF is defined using the **ipv6 max-mroute** command.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# ipv6 max-mroute
```

Syntax: [**no**] **ipv6 max-mroute** <*num*>

The <*num*> parameter specifies the maximum number of IPv6 multicast routes. If not defined by this command, the maximum value is determined by available system resources.

To define the maximum number of IPv6 multicast routes for a specified VRF, use the following commands.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 max-mroute
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

The **vrf** parameter specified with the **IPv6 router pim** command allows you to configure the **ipv6 max-mroute** command for a virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Defining the maximum number of IPv6 PIM cache entries

You can use the **max-mcache** command to define the maximum number of repeated PIM traffic being sent from the same source address and being received by the same destination address. To define the maximum for the default VRF, enter the **max-mcache** command.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# max-mcache 999
```

Syntax: [**no**] **max-mcache** <*num*>

The <*num*> variable specifies the maximum number of IPv6 multicast cache entries for PIM in the default VRF. The maximum value and the default value that can be entered is 4K. If not defined by this command, the maximum value is determined by available system resources.

To define the maximum number of IPv6 PIM Cache entries for a specified VRF, use the following command.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# max-mcache 999
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

The **vrf** parameter specified with the **IPv6 router pim** command allows you to configure the **max-mcache** command for a virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Defining the maximum number of IPv6 multicast VRF CAM entries for all VRFs

You can use the following run-time command to define the maximum number of IPv6 multicast VRF CAM entries for all non-default VRF instances by entering a command such as the following.

```
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 multicast-max-all-vrf-cam 3072
```

Syntax: [**no**] **ipv6 multicast-max-all-vrf-cam** <*cam-size*>

The **ipv6 multicast-max-all-vrf-cam** command is configured at the global level. The <*cam-size*> variable specifies the maximum number of multicast VRF CAM entries for all non-default VRF instances. This setting does not affect the default VRF. The maximum possible value is 8000 and the default value is 2048.

Defining the maximum number of IPv6 multicast VRF CAM entries for a specified VRF

You can use the following run-time command to define the maximum number of IPv6 multicast VRF CAM entries for a specified non-default VRF by entering commands such as the following.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 multicast-max-cam 3072
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

Syntax: [**no**] **ipv6 multicast-max-cam** <*cam-size*>

The **vrf** parameter specified with the **IPv6 router pim** command allows you to configure the **ipv6 multicast-max-cam** command for a virtual routing instance (VRF) specified by the variable <*vrf-name*>.

The <*cam-size*> variable specifies the maximum number of IPv6 multicast VRF CAM entries for one or more non-default specified VRFs. This setting can be any number up to the limit set using the **ipv6 multicast-max-all-vrf-cam** command.

PIM Anycast RP

PIM Anycast RP is a method of providing load balancing and fast convergence to PIM RPs in an IPv6 multicast domain. The RP address of the Anycast RP is a shared address used among multiple PIM routers, known as PIM RP. The PIM RP routers create an Anycast RP set. Each router in the Anycast RP set is configured using two IP addresses: a shared RP address in their loopback address and a separate, unique IP address. The loopback address must be reachable by all PIM routers in the multicast domain. The separate, unique IP address is configured to establish static peering with other PIM routers and communication with the peers.

When the source is activated in a PIM Anycast RP domain, the PIM First Hop (FH) will register the source to the closet PIM RP. The PIM RP follows the same MSDP Anycast RP operation by decapsulating the packet and creating the (s,g) state. If there are external peers in the Anycast RP set, the router will re-encapsulate the packet with the local peering address as the source address of the encapsulation. The router will unicast the packet to all Anycast RP peers. The re-encapsulation of the data register packet to Anycast RP peers ensures source state distribution to all RPs in a multicast domain.

Configuring PIM Anycast RP

A new PIM CLI is introduced for PIM Anycast RP under the router pim submode. The PIM CLI specifies mapping of the RP and the Anycast RP peers.

To configure PIM Anycast RP, enter the following commands.

```
NetIron(config)# ipv6 router pim
NetIron(config-ipv6-pim-router)# rp-address 1001::1
NetIron(config-ipv6-pim-router)# anycast-rp 1001::1 my-anycast-rp-set-acl
```

To configure PIM Anycast RP for a specified VRF, enter the commands as shown in the following example.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# rp-address 1001::1
NetIron(config-ipv6-pim-router-vrf-blue)# anycast-rp 1001::1
my-anycast-rp-set-acl
```

Syntax: [**no**] **anycast-rp** <*rp-address*> <*my-anycast-rp-set-acl*>

The <*rp address*> parameter specifies a shared RP address used among multiple PIM routers.

The <*my-anycast-rp-set-acl*> parameter specifies a host-based simple ACL used to specify the address of the Anycast RP set, including a local address.

The following example is a configuration of PIM Anycast RP 1001:1.The example avoids using the loopback 1 interface when configuring PIM Anycast RP because the loopback 1 address could be used as a router-id. A PIM First Hop router will register the source with the closest RP. The first RP that receives the register will re-encapsulate the register to all other Anycast RP peers. Refer to Figure 221 as described in the configuration of PIM Anycast RP 1001:1.

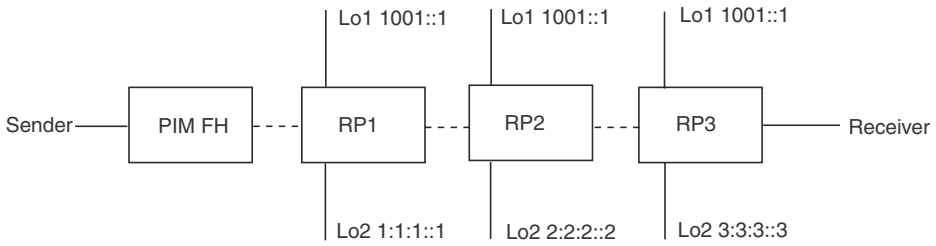
```
NetIron(config)# interface loopback 2
NetIron(config-lbif-2)# ipv address 1001::1/96
NetIron(config-lbif-2)# ipv pim-sparse
NetIron(config-lbif-2)# interface loopback 3
NetIron(config-lbif-3)# ipv address 1:1:1::1/96
NetIron(config-lbif-3)# ipv pim-sparse
NetIron(config-lbif-3)# ipv6 router pim
NetIron(config-ipv6-pim-router)# rp-address 1001::1
NetIron(config-ipv6-pim-router)# anycast-rp 1001::1 my-anycast-rp-set
NetIron(config-ipv6-pim-router)# ipv6 access-list my-anycast-rp-set
NetIron(config-std-nacl)# permit ipv6 host 1:1:1::1 any
NetIron(config-std-nacl)# permit ipv6 host 2:2:2::2 any
NetIron(config-std-nacl)# permit ipv6 host 3:3:3::3 any
```

The RP shared address 1001:1 is used in the PIM domain. IP addresses 1:1:1:1, 2:2:2:2, and 3:3:3:3 are listed in the ACL that forms the self-inclusive Anycast RP set. Multiple Anycast RP instances can be configured on a system; each peer with the same or different Anycast RP set.

NOTE
The PIM Anycast CLI applies to only PIM routers running RP. All deny statements in the anycast_rp_set ACL are ignored.

The example shown in Figure 221 is a PIM Anycast-enabled network with three RPs and one PIM-FH router connecting to its active source and local receiver. Loopback 2 in RP1, RP2, and RP3 each have the same IP addresses 1001:1. Loopback 3 in RP1, RP2, and RP3 each have separate IP address configured to communicate with their peers in the Anycast RP set.

FIGURE 221 Example of a PIM Anycast RP network



Displaying information for an IPv6 PIM Anycast RP interface

To display information for an IPv6 PIM Anycast RP interface, enter the **show ipv6 pim anycast-rp** command.

```
NetIron(config)# show ipv6 pim anycast-rp
Number of Anycast RP: 1
Anycast RP: 1001::1
  ACL ID: 200
  ACL Name: my-anycast-rp-set
  ACL Filter: SET
  Peer List:
   1:1:1::1
   2:2:2::2
   3:3:3::3
```

Syntax: show ipv6 pim [**vrf** <*vrf-name*>] <*anycast-rp*>

The **vrf** parameter allows you to display information for an IPv6 Anycast RP interface for the VRF instance identified by the <*vrf-name*> variable.

Table 380 describes the parameters of the **show ipv6 pim anycast-rp** command.

TABLE 380 Output from the **show ipv6 pim anycast-rp** command

Field	Description
Number of Anycast RP	Specifies the number of Anycast RP sets in the multicast domain.
Anycast RP	Specifies a shared RP address used among multiple PIM routers.
ACL ID	Specifies the ACL ID assigned.
ACL Name	Specifies the name of the Anycast RP set.
ACL Filter	Specifies the ACL filter state SET or UNSET.
Peer List	Specifies host addresses that are permitted in the Anycast RP set.

Multicast Listener Discovery and source-specific multicast protocols

Multicast Listener Discovery Version 2 (MLDv2) protocol is supported. IPv6 routers use the MLDv2 protocol to discover multicast listeners, or nodes that wish to receive multicast packets on directly attached links. MLDv2 supports source filtering, the ability of a node to send reports on traffic that is from a specific address source or from all multicast addresses except the specified address sources. The information is then provided to the source-specific multicast (SSM) routing protocols such as PIM-SSM.

The IPv6 router stores a list of multicast addresses for each attached link. For each multicast address, the IPv6 router stores a filter mode and a source list. The filter mode is set to INCLUDE if all nodes in the source list for a multicast address are in the INCLUDE state. If the filter mode is INCLUDE, then only traffic from the addresses in the source list is allowed. The filter mode is set to EXCLUDE if at least one of the nodes in the source list is in an EXCLUDE state. If the filter mode is EXCLUDE, traffic from nodes in the source list is denied and traffic from other sources is allowed.

The source list and filter mode are created when the IPv6 querier router sends a query. The querier router is the one with the lowest source IPv6 address. It sends out any of the following queries:

- **General query** – The querier sends this query to learn all multicast addresses that need to be listened to on an interface.

- **Address specific query** – The querier sends this query to determine if a specific multicast address has any listeners.
- **Address specific and source specific query** – The querier sends this query to determine if specified sources of a specific multicast address have any listeners.

In response to these queries, multicast listeners send the following reports:

- **Current state** – This report specifies the source list for a multicast address and whether the filter mode for that source list is INCLUDE or EXCLUDE.
- **Filter-mode change** – This report specifies if there has been a change to the filter mode for the source list and provides a new source list.
- **Source list change** – This report specifies the changes to the source list.

MLDv1 is compatible with IGMPv2 and MLDv2 is compatible with IGMPv3.

Enabling MLDv2

MLDv1 is enabled once PIM Sparse Mode (PIM-SM) is enabled on an interface. You then enable version 2 of MLD, the version that supports source filtering.

MLDv2 interoperates with MLDv1. MLDv1 messages are understood by MLDv2. When an IPv6 router detects that the node is operating in MLDv1 mode, the router switches to MLDv1 for that node even though queries are sent in MLDv2.

To enable IPv6 PIM-SM, enter the following command at the interface level.

```
NetIron(config)# ipv6 router pim
NetIron(config-if-e10000-1/1)# ipv6 pim-sparse
```

Syntax: [no] **ipv6 pim-sparse**

Configuring MLD parameters for default and non-default VRFs

MLD allows you to configure the following parameters on default and non-default VRFs:

- Group membership time - "Setting the group membership time" on page 1982
- Max group address - "Defining the maximum number of MLD group addresses" on page 1983
- Max response time - "Setting the maximum response time" on page 1983
- Query interval - "Setting the query interval" on page 1984
- Last listener query count - "Setting the last listener query interval" on page 1984
- Last listener query interval - "Setting the last listener query interval" on page 1984
- Robustness - "Setting the robustness" on page 1984
- Version - "Setting the version" on page 1985

Setting the group membership time

You can set the group membership time for the default VRF or for a specified VRF. Group membership time defines how long a group will remain active on an interface in the absence of a group report. Possible values are from 5 through 26,000 seconds and the default value is 260 seconds.

To define an MLD group membership time of 2000 seconds, enter the following command.

```
NetIron(config)# ipv6 mld group-membership-time 2000
```

Syntax: [**no**] **ipv6 mld group-membership-time** <5-26000>

To define an MLD group membership time of 2000 seconds for a specified VRF, enter the following commands.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 mld group-membership-time 2000
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

The **vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Defining the maximum number of MLD group addresses

You can use the following run-time command to set the maximum number of MLD addresses for the default VRF or for a specified VRF. To define this maximum for the default VRF, enter the following command.

```
NetIron(config)# ipv6 mld max-group-address 1000
```

Syntax: [**no**] **ipv6 mld max-group-address** <*num*>

The <*num*> variable specifies the maximum number of MLD group addresses you want to make available for the default VRF. If not defined by this command, the maximum value is determined by available system resources.

To define this maximum for a specified VRF, enter the following commands.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 mld max-group-address 1000
```

Syntax: [**no**] **ipv6 router pim vrf** [<*vrf-name*>]

The **vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Setting the maximum response time

You can define the maximum amount of time a multicast listener has to respond to queries by entering a command such as the following.

```
NetIron(config)# ipv6 mld max-response-time 5
```

Syntax: [**no**] **ipv6 mld max-response-time** <*seconds*>

The <*seconds*> variable specifies the MLD maximum response time in seconds. You can specify from 1 through 10 seconds. The default is 5 seconds.

To define the maximum amount of time a multicast listener has to respond to queries for a specified VRF, enter the following commands.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 mld max-response-time 5
```

Syntax: [**no**] **ipv6 router pim vrf** [<*vrf-name*>]

The **vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Setting the query interval

You can define the frequency at which MLD query messages are sent. For example, if you want queries to be sent every 50 seconds, enter a command such as the following.

```
NetIron(config)# ipv6 mld query-interval 50
```

Syntax: [**no**] **ipv6 mld query-interval** <*seconds*>

The <*seconds*> variable specifies the MLD query interval in seconds. You can specify from 1 through 3600 seconds. The default value is 125 seconds.

To define the frequency at which MLD query messages are sent for a specified VRF, enter the following commands.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 mld query-interval 50
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

The **vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Setting the last listener query interval

The Last Listener Query Interval is the Maximum Response Delay inserted into Multicast-Address-Specific Queries sent in response to Done messages, and is also the amount of time between Multicast- Address-Specific Query messages. When the device receives an MLDv1 leave message or an MLDv2 state change report, it sends out a query and expects a response within the time specified by this value. Using a lower value allows members to leave groups more quickly. You can set the last listener query interval by entering a command such as the following.

```
NetIron(config)# ipv6 mld llqi 5
```

Syntax: [**no**] **ipv6 mld llqi** <*seconds*>

The <*seconds*> variable sets the last listener query interval in seconds. You can specify from 1 through 10 seconds.

To set the last listener query interval for a specified VRF, enter the following commands.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 mld llqi 5
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

The **vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Setting the robustness

You can specify the number of times that the switch sends each MLD message from this interface. Use a higher value to ensure high reliability from MLD. You can set the robustness by entering a command such as the following.

```
NetIron(config)# ipv6 mld robustness 3
```

Syntax: ipv6 mld robustness <*seconds*>

The <*seconds*> variable sets the MLD robustness in seconds. You can specify from 2 through 7 seconds. The default is 2 seconds.

To set the robustness for a specified VRF, enter the following commands.

```
NetIron(config)# ipv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 mld robustness 3
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

The **vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Setting the version

You can use this command to set the MLD version (1 or 2) globally. You can select the version of MLD by entering a command such as the following.

```
NetIron(config)# ipv6 mld version 2
```

Syntax: **ipv6 mld version** <*version-number*>

The <*version-number*> variable sets the MLD version. You can specify 1 or 2 for the MLD version.The default version is 2.

To set the robustness for a specified VRF, enter the following commands.

```
NetIron(config)# pv6 router pim vrf blue
NetIron(config-ipv6-pim-router-vrf-blue)# ipv6 mld version 2
```

Syntax: [**no**] **ipv6 router pim** [**vrf** <*vrf-name*>]

The **vrf** parameter specifies the virtual routing instance (VRF) specified by the variable <*vrf-name*>.

Configuring MLD parameters at the interface level

The following MLD parameters can be configured at the interface level:

- Port- version - "Specifying a port version" on page 1985
- Static-group - "Specifying a static group" on page 1985
- Tracking - "Enabling MLD tracking on an interface" on page 1986
- Version - "Setting the version on an interface" on page 1986

Specifying a port version

To set the MLD version on a virtual Ethernet interface, enter the following commands as shown in the example.

```
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ipv6 mld port-version 2
```

Syntax: **ipv6 mld port-version** <*version-number*>

Enter 1 or 2 for <*version-number*>. Be sure to enter 2 if you want to use source filtering.

Specifying a static group

A multicast group is usually learned when an MLDv1 report is received. You can configure static group membership without having to receive an MLDv1 report by entering a command such as the following at the interface level.

```
NetIron(config-if-e10000-1/1)# ipv6 mld static-group ff0d::1
```

To configure a static group membership without having to receive an MLDv1 report on a virtual Ethernet interface, enter the following commands.

```
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ipv6 mld static-group ff0d::1
```

Syntax: **ipv6 mld static-group** <*multicast-group-address*> [**ethernet** <*port-number*> [**ethernet** <*port-number*> | **to** <*port-number*>]*]

Enter the IPv6 multicast group address for the <*multicast-group-address*>.

Enter the number of the port that will be included in this static group for the **ethernet** <*port-number*> parameter. The asterisk (*) in the syntax means that you can enter as many port numbers as you want to include in the static group. For a virtual routing interface (ve), specify the physical Ethernet ports on which to add the group address.

Enabling MLD tracking on an interface

When MLD tracking is enabled, a Layer 3 switch tracks all clients that send membership reports. When a Leave message is received from the last client, the device immediately stops forwarding to the physical port (without waiting 3 seconds to confirm that no other clients still want the traffic). To enable MLD tracking on a virtual interface, enter the following commands.

```
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ipv6 mld tracking
```

Syntax: **ipv6 mld tracking**

Setting the version on an interface

You can use this command to set the MLD version (1 or 2) on an interface. You can select the version of MLD by entering a command such as the following.

```
NetIron(config)# interface ve 10
NetIron(config-vif-10)# ipv6 mld version 2
```

Syntax: **ipv6 mld version** <*version-number*>

The <*version-number*> variable sets the MLD version on an interface. You can specify 1 or 2 for the MLD version.The default version is 2.

Displaying MLD information

The sections below present the show commands for MLD.

Displaying MLD group information

To display the list of multicast groups, enter a command such as the following.

```
NetIron #show ipv6 mld group
Interface e6/18 has 11 groups
    group                                phy-port static querier life mode

1    ff33::6:b:1                          e6/18    no     yes     0    incl
2    ff33::6:a:1                          e6/18    no     yes     0    incl
3    ff33::6:9:1                          e6/18    no     yes     0    incl
4    ff33::6:8:1                          e6/18    no     yes     0    incl
5    ff33::6:7:1                          e6/18    no     yes     0    incl
6    ff33::6:6:1                          e6/18    no     yes     0    incl
7    ff33::6:5:1                          e6/18    no     yes     0    incl
8    ff33::6:4:1                          e6/18    no     yes     0    incl
9    ff33::6:3:1                          e6/18    no     yes     0    incl
10   ff33::6:2:1                          e6/18    no     yes     0    incl
11   ff33::6:1:1                          e6/18    no     yes     0    incl
```

Syntax: show ipv6 mld [**vrf** <*vrf-name*>] **group**

The **vrf** parameter allows you to display the list of IPv6 MLD groups for the VRF instance identified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 mld group** command.

TABLE 381 Output from the **show ipv6 mld group** command

Field	Description
Interface <*port-number*> has x groups	This message shows the ID of the interface and how many multicast groups it has.
#	Index for the MLD group.
group	IPv6 address of the multicast group.
phy-port	The physical port to which the group belongs.
static	Indicates if the group is a static group or not.
querier	Indicates if the multicast group is a querier or not.
life	The number of seconds the interface can remain in its current mode.
mode	Indicates if the filter mode of the multicast group is in INCLUDE or EXCLUDE.

Displaying MLD definitions for an interface

To display the MLD parameters on an interface, including the various timers, the current querying router, and whether or not MLD is enabled, enter the following command.

```
NetIron# show ipv6 mld interface
version = 2, query int = 60, max resp time = 5, group mem time = 140
e3/1: default V2, PIM sparse, addr=fe80::20c:dbff:fe82:833a
e3/2: default V2, PIM sparse, addr=fe80::20c:dbff:fe82:833b
e6/1: default V2, PIM sparse (port down), addr=::
e6/5: default V2, PIM sparse (port down), addr=::
e6/18: default V2, PIM sparse, addr=fe80::20c:dbff:fe82:840b
          has   11 groups, Querier, default V2
    group: ff33::6:b:1, include, permit 1
    group: ff33::6:a:1, include, permit 1
    group: ff33::6:9:1, include, permit 1
    group: ff33::6:8:1, include, permit 1
    group: ff33::6:7:1, include, permit 1
    group: ff33::6:6:1, include, permit 1
    group: ff33::6:5:1, include, permit 1
    group: ff33::6:4:1, include, permit 1
    group: ff33::6:3:1, include, permit 1
    group: ff33::6:2:1, include, permit 1
    group: ff33::6:1:1, include, permit 1
```

Syntax: show ipv6 mld [**vrf** <*vrf-name*>] **interface** [<*port-number*>]

The **vrf** parameter allows you to display MLD parameters on an interface for the VRF instance identified by the <*vrf-name*> variable.

Enter a port number in the <*port-number*> variable if you want to display MLD information for a specific interface.

Table displays the output from the **show ipv6 mld interface** command.

TABLE 382 Output from the **show ipv6 mld interface** command

Field	Description
version	Version of the MLD being used.
query int	Query interval in seconds.
max resp time	Number of seconds multicast groups have to respond to queries.
group mem time	Number of seconds multicast groups can be members of this group before aging out.
(details)	The following is displayed for each interface: • The port ID • The default MLD version being used • The multicast protocol used • IPV6 address of the multicast interface • If the interface has groups, the group source list, IPv6 multicast address, and the filter mode are displayed.

To display the MLD parameters on an interface for a specified VRF, enter the following command as shown in the example below.

```
NetIron(config)# show ipv mld vrf public interface
---------+------+---------+-----------------------------------------+---------+---
--+---------
Intf/Port|Groups| Version |Querier                                  |  Timer
|V1Rtr|Tracking
         |      | Oper  Cfg|                                         | OQrr GenQ|
|
---------+------+----+----+-----------------------------------------+----+----+---
--+---------
v6             0    2    -
Disabled
     e5/1           2    - fe80::20c:dbff:fee2:5000                 11    0 No
v61            0    2    -
Disabled
    e11/1           2    - Self                                      0  122 No
```

Displaying MLD settings

To display MLD settings for the “eng” VRF, enter the following command.

```
NetIron# show ipv6 mld vrf eng settings
MLD Global Configuration
  Query Interval            : 125s   Configured Interval    : 125s
  Max Response Time         : 10s
  Group Membership Time     : 260s
  Operating Version         : 2      Configured Version     : 0
  Robustness Variable       : 2
  Last Member Query Interval: 1s      Last Member Query Count: 2
  Older Host Present Timer  : 260s
```

Syntax: show ipv6 mld [**vrf** <*vrf-name*>] **settings**

The **vrf** parameter specifies that you want to display information for MLD settings for the VRF specified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 mld vrf eng settings** command.

TABLE 383 Output from the **show ipv6 mld vrf eng settings** command

Field	Description
Query Interval	How often the router will query an interface for group membership.
Configured Interval	The interval that has been configured for the router.
Max Response Time	The length of time in seconds that the router will wait for an IGMP (V1 or V2) response from an interface before concluding that the group member on that interface is down and removing it from the group.
Group Membership Time	The length of time in seconds that a group will remain active on an interface in the absence of a group report.
Operating Version	The IGMP version operating on the router.
Configured Version	The IGMP version configured on the router.
Robustness Variable	Used to fine-tune for unexpected loss on the subnet. The value is used to calculate the group interval.

TABLE 383 Output from the **show ipv6 mld vrf eng settings** command (Continued)

Field	Description
Last Member Query Interval	Indicates when a leave is received; a group-specific query is sent. The last member query count is the number of queries with a time interval of (LMQT) is sent.
Last Member Query Count	Specifies the number of group-specific queries when a leave is received.

Displaying static MLD groups

The following command displays static MLD groups for the “cs” VRF.

```
NetIron# show ipv6 mld vrf cs static
Group Address                                Interface Port List
-----------------------------------------+---------+---------
ff1e:1::1                                    v3       ethe 2/10
ff1e:a::7f                                   v3       ethe 2/10
```

Syntax: show ipv6 mld [**vrf** <*vrf-name*>] **static**

The **vrf** parameter specifies that you want to display static MLD group information for the VRF specified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 mld vrf cs static** command.

TABLE 384 Output from the show ipv6 mld vrf cs static command

Field	Description
Group Address	The address of the multicast group.
Interface Port List	The physical ports on which the multicast groups are received.

Displaying MLD traffic

To display information on MLD traffic, enter a command such as the following.

```
NetIron# show ipv6 mld traffic
Recv  QryV1 QryV2 G-Qry GSQry MbrV1 MbrV2 Leave IS_IN IS_EX 2_IN 2_EX ALLO  BLK
e3/1      0     0     0     0     0     0     0     0     0    0    0    0    0
e3/2      0     0     0     0     0     0     0     0     0    0    0    0    0
e6/18     0     0     0     0     0   176     0   110     0    0    0   66    0
e6/19     0     0     0     0     0   176     0   110     0    0    0   66    0
e6/20     0     0     0     0     0   176     0   110     0    0    0   66    0
e6/25     0     0     0     0     0   176     0   110     0    0    0   66    0
l1        0     0     0     0     0     0     0     0     0    0    0    0    0
Send  QryV1 QryV2 G-Qry GSQry
e3/1      0     0     0     0
e3/2      0     0     0     0
e6/18     0    10    10     0
e6/19     0    10    10     0
e6/20     0    10    10     0
e6/25     0    10    10     0
l1        0     0     0     0
R2#
```

The report has a Receive and a Send section.

Syntax: show ipv6 mld [**vrf** <*vrf-name*>] **traffic**

The **vrf** parameter specifies that you want to display information on MLD traffic for the VRF specified by the <*vrf-name*> variable.

Table displays the output from the **show ipv6 mld traffic** command.

TABLE 385 Output from the **show ipv6 mld traffic** command

Field	Description
QryV1	Number of general MLDv1 queries received or sent by the virtual routing interface.
QryV2	Number of general MLDv2 queries received or sent by the virtual routing interface.
G-Qry	Number of group-specific queries received or sent by the virtual routing interface.
GSQry	Number of source specific queries received or sent by the virtual routing interface.
MbrV1	Number of MLDv1 membership reports received.
MbrV2	Number of MLDv2 membership reports received.
Leave	Number of MLDv1 “leave” messages on the interface. (See 2_Ex for MLDv2.)
Is_IN	Number of source addresses that were included in the traffic.
Is_EX	Number of source addresses that were excluded in the traffic.
2_IN	Number of times the interface mode changed from exclude to include.
2_EX	Number of times the interface mode changed from include to exclude.
ALLOW	Number of times that additional source addresses were allowed or denied on the interface.
BLK	Number of times that sources were removed from an interface.

Clearing IPv6 MLD traffic

To clear counters on IPv6 MLD traffic, enter the following command.

```
NetIron# clear ipv6 mld traffic
```

Syntax: clear ipv6 mld [**vrf** <*vrf-name*>] **traffic**

Use the **vrf** option to clear counters on IPv6 MLD traffic for a VRF instance specified by the <*vrf-name*> variable.

Clearing the IPv6 MLD group membership table cache

You can clear the IPv6 PIM group membership table cache using the following command.

```
NetIron# clear ipv6 pim cache
```

Syntax: clear ipv6 pim [**vrf** <*vrf-name*>] **cache**

Use the **vrf** option to clear the IPv6 PIM group membership table cache for a VRF instance specified by the <*vrf-name*> variable.

IPv6 Multicast Listener Discovery snooping

IPv6 Multicast Listener Discovery (MLD) snooping controls the amount of multicast traffic in a switched network. By default, a LAN switch floods the broadcast domain with multicast IPv6 packets. If many multicast servers are sending streams to the segment, this will consume a lot of bandwidth. MLD snooping identifies multicast-enabled router ports and multicast receiver ports in a given VLAN or a switched network and forwards multicast traffic only to those ports.

Configuring IPv6 multicast routing or snooping

IPv6 multicast snooping or routing can be enabled on a VE interface or VLAN, but not on both. This is because all of the multicast data and control packets received on the snooping VLAN are handled by multicast snooping and do not reach the multicast routing component. Similarly, any multicast data or control packets received on a VE interface enabled with PIM or Distance Vector Multicast Routing Protocol (DVMRP) routing are handled by the PIM or DVMRP routing component and are not seen by the MLD snooping component.

The following considerations apply when configuring concurrent operation of multicast routing and snooping.

- Either multicast snooping or routing can be enabled on a VE or VLAN but not both.
- Snooping can be enabled globally (**ipv6 multicast active** | **passive**).
- The global snooping configuration is inherited by all current VLANs that are not enabled for multicast routing.
- The global snooping configuration is also inherited by all new VLANs. To enable multicast routing on a newly configured VE or VLAN (when snooping is globally enabled), you must first disable snooping on the newly created VE or VLAN.
- Global snooping configuration must be configured first before VLAN configuration.
- A VLAN-level snooping configuration is displayed only if it is different from the global configuration.

Enabling IPv6 multicast traffic reduction

By default, the device forwards all IPv6 multicast traffic out to all ports except the port on which the traffic was received. To reduce multicast traffic through the device, you can enable IPv6 Multicast Traffic Reduction. This feature configures the device to forward multicast traffic only on the ports attached to multicast group members, instead of forwarding all multicast traffic to all ports. The device determines the ports that are attached to multicast group members based on entries in the MLD Snooping table. Each entry in the table consists of MAC addresses and the ports from which the device has received Group Membership reports for that group.

By default, the device broadcasts traffic addressed to an IPv6 multicast group that does not have any entries in the MLD Snooping table. When you enable IPv6 Multicast Traffic Reduction, the device determines the ports that are attached to multicast group members based on entries in the MLD Snooping table. The MLD Snooping table entries are created when the VLAN receives a Group Membership report for a group. Each entry in the table consists of an IPv6 multicast group address and the ports from which the device has received Group Membership reports.

When the device receives traffic for an IPv6 multicast group, the device looks in the MLD Snooping table for an entry corresponding to that group. If the device finds an entry, the device forwards the group traffic out to the ports listed in the corresponding entries, as long as the ports are members of the same VLAN. If the table does not contain an entry corresponding to the group or if the port is a member of the default VLAN, the device broadcasts the traffic.

When one or more devices are running Layer 2 IPv6 Multicast Traffic Reduction, configure one of the devices for active MLD and leave the other devices configured for passive MLD. However, if the IPv6 multicast domain contains a multicast-capable router, configure all the devices for MLD and allow the router to actively send the MLD queries.

Configuring IPv6 MLD snooping

To enable IPv6 Multicast Traffic Reduction, enter the following command.

```
NetIron(config)# ipv6 multicast active
```

Syntax: [no] ipv6 multicast active | passive

When you enable IPV6 multicast on a device, all ports on the device are configured for MLD.

If you are using passive MLD, all ports can send MLD queries and receive MLD reports. If you are using passive MLD, all ports can receive MLD queries.

IPv6 Multicast Traffic Reduction cannot be disabled on individual ports of a device. IPv6 Multicast Traffic Reduction can be disabled globally by entering the no ipv6 multicast command.

To verify that IPv6 Multicast Traffic Reduction is enabled, enter the following command at any level of the CLI.

```
NetIron(config)# show ipv6 multicast
IPv6 multicast is enabled - Active
```

Syntax: show ipv6 multicast

Changing the MLD mode

When you enable IPv6 Multicast Traffic Reduction on the device, MLD also is enabled. The device uses MLD to maintain a table of the Group Membership reports received by the device. You can use active or passive MLD mode. There is no default mode.

The active and passive MLD modes are described as follows:

- Active – When active MLD mode is enabled, the device actively sends out MLD queries to identify IPV6 multicast groups on the network and makes entries in the MLD table based on the Group Membership reports received from the network.

  NOTE
  Routers in the network generally handle this operation. Use the active MLD mode only when the device is in a standalone Layer 2 switched network with no external IPV6 multicast router attachments. In this case, enable the active MLD mode on only one of the devices and leave the other devices configured for passive MLD mode.

- Passive – When passive MLD mode is enabled, the device listens for MLD Group Membership reports but does not send MLD queries. The passive mode is sometimes called “MLD snooping”. Use this mode when another device in the network is actively sending queries.

Globally configuring the age interval

When the device receives a Group Membership report, the device makes an entry in the MLD group table for the group in the report. The age interval specifies how long the entry can remain in the table without the device receiving another Group Membership report.

To configure the age interval, enter a command such as the following.

```
NetIron(config)# ipv6 multicast age-interval 280
```

Syntax: [no] ipv6 multicast age-interval <*interval*>

The <*interval*> parameter specifies the interval between queries. You can specify a value from 10 through 1220 seconds. The default is 260 seconds.

Filtering multicast groups

By default, the device forwards multicast traffic for all valid multicast groups. You can configure a device to filter out all multicast traffic for groups other than the ones for which the device has received Group Membership reports.

When the device starts up, it forwards all multicast groups even though multicast traffic filters are configured. This process continues until the device receives a Group Membership report. Once the Group Membership report is received, the device drops all multicast packets for groups other than the ones for which the device has received the Group Membership report.

To enable IPv6 multicast filtering, enter the following command.

```
NetIron(config)# ipv6 multicast filter
```

Syntax: [no] ipv6 multicast filter

Setting PIM proxy interval

The PIM proxy interval specifies the time interval in seconds between the PIM proxy and join messages.

To set the time interval between PIM proxy messages, enter the following command.

```
NetIron(config)# ipv6 multicast pim-proxy-interval 10
```

Syntax: ipv6 multicast pim-proxy-interval <*interval*>

The <*interval*> parameter specifies the interval between PIM proxy messages. You can specify a value from 10 through 600 seconds.

PIM-SM traffic snooping

By default, when a device receives an IPv6 multicast packet, the device does not examine the multicast information in the packet. Instead, the device simply forwards the packet out to all ports except the port that received the packet. In some networks, this method can cause unnecessary traffic overhead in the network. For example, if the device is attached to only one group source and two group receivers, but has devices attached to every port, the device forwards group traffic out all ports in the same broadcast domain except the port attached to the source, even though there are only two receivers for the group.

PIM-SM traffic snooping eliminates the superfluous traffic by configuring the device to forward IPv6 multicast group traffic only on the ports that are attached to receivers for the group.

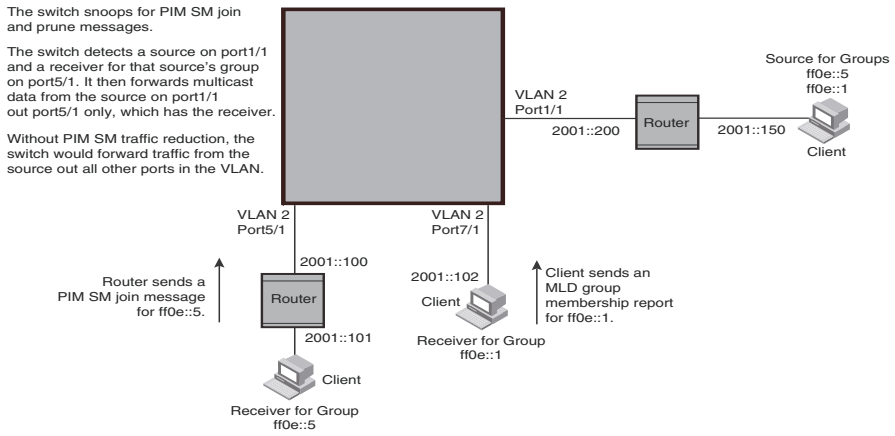
PIM-SM traffic snooping requires IPv6 Multicast Traffic Reduction to be enabled on the device. IPv6 Multicast Traffic Reduction configures the device to listen for MLD messages. PIM-SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for PIM-SM join and prune messages sent from one PIM-SM router to another through the device.

Application examples

Figure 222 shows an example application of the PIM-SM traffic snooping feature. In this example, a device is connected through an IP router to a PIM-SM group source that is sending traffic for two PIM-SM groups. The device also is connected to a receiver for each of the groups.

NOTE
PIM-SM traffic snooping applies only to PIM-SM version 2 (PIM V2).

FIGURE 222 PIM-SM IPv6 traffic reduction in enterprise network



When PIM-SM traffic snooping is enabled, the device starts listening for PIM-SM join and prune messages and MLD Snooping reports. Until the device receives a PIM-SM join message or an MLD report, the device forwards IPv6 multicast traffic out to all ports. Once the device receives a join message or Group Membership report for a group, the device forwards subsequent traffic for that group only on the ports from which the join messages or MLD reports were received.

In this example, the router connected to the receiver for group ff0e::1 sends a join message toward the source of the group. Since PIM-SM traffic snooping is enabled on the device, the device examines the join message to learn the group ID, then makes a forwarding entry for the group ID and the port connected to the router of the receiver. The next time the device receives traffic for ff0e::1 from the source of the group, the device forwards the traffic only on port 5/1, because that is the only port connected to a receiver for the group.

Notice that the receiver for group ff0e::5 is directly connected to the device. As a result, the device does not see a join message on behalf of the client. However, because IP Multicast Traffic Reduction also is enabled, the device uses the MLD Group Membership report from the client to select the port for forwarding traffic to group ff0e::5 receivers.

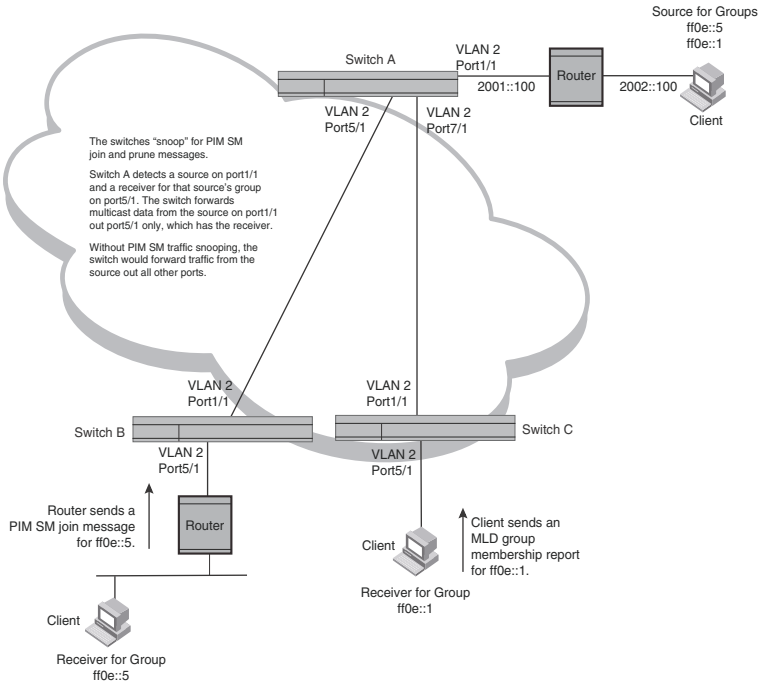
The IPv6 Multicast Traffic Reduction feature and the PIM-SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM-SM groups learned through join messages as well as MAC addresses learned through MLD reports. In this case, even though the device never sees a join message for the receiver for group ff0e::5, the device nonetheless learns about the receiver and forwards group traffic to the receiver.

The device stops forwarding IPv6 multicast traffic on a port for a group if the port receives a prune message for the group.

Notice that the ports connected to the source and the receivers are all in the same port-based VLAN on the device. This is required for the PIM-SM traffic snooping feature. The feature also requires the source and the downstream router to be on different IP subnets, as shown in Figure 222.

Figure 223 shows another example application for PIM-SM traffic snooping. This example shows devices on the edge of a global Ethernet cloud (a Layer 2 Packet over SONET cloud). Assume that each device is attached to numerous other devices.

FIGURE 223 PIM-SM IPv6 traffic reduction in global Ethernet environment



The devices on the edge of the global Ethernet cloud are configured for IP Multicast Traffic Reduction and PIM-SM traffic snooping. Although this application uses multiple devices, the feature has the same requirements and works the same way as it does on a single device.

Configuration requirements

Consider the following configuration requirements:

- IPv6 Multicast Traffic Reduction must be enabled on the device that will be running PIM-SM snooping. The PIM-SM traffic snooping feature requires IPv6 Multicast Traffic Reduction.

  NOTE
  Use the passive mode of IPv6 Multicast Traffic Reduction instead of the active mode. The passive mode assumes that a router is sending group membership queries as well as join and prune messages on behalf of receivers. The active mode configures the device to send group membership queries.

- All the device ports connected to the source and receivers or routers must be in the same port-based VLAN.
- The PIM-SM snooping feature assumes that the group source and the device are in different subnets and communicate through a router. The source must be in a different IP subnet than the receivers. A PIM-SM router sends PIM join and prune messages on behalf of a multicast group receiver only when the router and the source are in different subnets. When the receiver and source are in the same subnet, they do not need the router in order to find one another. They find one another directly within the subnet.

  The device forwards all IPv6 multicast traffic by default. Once you enable IPv6 Multicast Traffic Reduction and PIM-SM traffic snooping, the device initially blocks all PIM-SM traffic instead of forwarding it. The device forwards PIM-SM traffic to a receiver only when the device receives a join message from the receiver. Consequently, if the source and the downstream router are in the same subnet, and PIM-SM traffic snooping is enabled, the device blocks the PIM-SM traffic and never starts forwarding the traffic. This is because the device never receives a join message from the downstream router for the group. The downstream router and group find each other without a join message because they are in the same subnet.

  NOTE
  If the “route-only” feature is enabled, PIM-SM traffic snooping is not supported.

Globally enabling IPv6 PIM SM traffic snooping

This feature is similar to PIM-SM traffic snooping but listens only for MLD snooping information, not PIM-SM information. You must enable both IPv6 Multicast Traffic Reduction and IPv6 PIM-SM traffic snooping to enable the device to listen for PIM-SM join and prune messages.

To enable IPv6 PIM-SM traffic snooping, enter the following commands at the global CONFIG level of the CLI.

```
NetIron(config)# ipv6 multicast active
NetIron(config)# ipv6 multicast pimsm-snooping
```

Syntax: [**no**] **ipv6 multicast** [**active** | **passive**]

When you enable IP multicast on device, all ports on the device are configured for IGMP.

If you are using **active** MLD, all ports can send MLD queries and receive MLD reports. If you are using **passive** MLD, all ports can receive MLD queries.

IPv6 Multicast Traffic Reduction cannot be disabled on individual ports of a device. IPv6 Multicast Traffic Reduction can be disabled globally by entering the **no ipv6 multicast** command.

Syntax: [**no**] **ipv6 multicast pimsm-snooping**

Use the **no** form of the command to disable IPv6 PIM-SM traffic snooping.

Globally configuring the query interval

If IPv6 Multicast Traffic Reduction is set to active mode, you can configure the query interval, which specifies how often a device is enabled for active IPv6 Multicast Traffic Reduction sends Group Membership queries.

NOTE
The query interval applies only to the active mode of IPv6 Multicast Traffic Reduction.

To configure the query interval, enter the following command.

```
NetIron(config)# ipv6 multicast query-interval 120
```

Syntax: [**no**] **ipv6 multicast query-interval** <*interval*>

The <*interval*> parameter specifies the interval between queries. You can specify a value from 10 through 600 seconds. The default is 125 seconds.

Setting the MLD version

You can use the **ipv6 multicast version** command to set the MLD version (1 or 2) globally for IPv6 multicast. You can select the version of MLD by entering the following command.

```
NetIron(config)# ipv6 multicast version 2
```

Syntax: **ipv6 multicast version** <*version-number*>

Enter 1 or 2 for the <*version-number*>. The default is version 1.

Configuring IPv6 multicast tracking and fast-leave

When IPv6 multicast tracking is configured, MLD fast-leave is enabled. MLD Leave Processing for MLDv1 is initiated by a receiver sending a Leave message. The router sends out a group-specific query to solicit reports from other receivers. The multicast group entry is maintained for 3 seconds to allow the processing of reports from any receivers on the LAN segment. If there are no receivers, the multicast stream is pruned.

MLD fast-leave allows a receiver to move from one multicast group to another instantly if it is the only receiver on the segment subscribed to the group. When a layer 3 switch receives a Leave message, it sends a group-specific query to see if any other receivers are present. The multicast group state is maintained for 3 seconds to process any MLD group reports from other receivers. If there are no other receivers, the multicast group entry prunes the receiver LAN segment, stopping traffic instantly.

MLDv2 also supports fast-leave processing. When an MLDv2 receiver changes the mode to Exclude, and there are no other receivers on that interface, the multicast group entry prunes the receiver LAN segment.

To enable IPv6 multicast tracking globally, enter the following command.

```
NetIron(config)# ipv6 multicast tracking
```

Syntax: [**no**] **ipv6 multicast tracking**

The **no** form of this command disables the tracking process globally.

Configuring IPv6 MLD snooping on a per-VLAN basis

The following IPv6 MLD snooping parameters can be configured on a per-VLAN basis:

- Active and passive - "Configuring IPv6 multicast snooping per VLAN" on page 1999
- MLD proxy - "Configuring MLD proxy per VLAN" on page 1999
- PIM-SM traffic snooping - "Configuring the PIM-SM traffic snooping per VLAN" on page 2000
- PIM proxy - "Configuring PIM proxy per VLAN" on page 2000
- Static-group uplink - "Configuring an IPv6 multicast static group uplink per VLAN" on page 2000
- Tracking - "Configuring IPv6 multicast tracking and fast-leave per VLAN" on page 2001

Configuring IPv6 multicast snooping per VLAN

The **multicast6** command allows you to configure IPv6 multicast snooping parameters per VLAN. To configure IPv6 multicast snooping to VLAN 2, enter the following commands as shown in the example below.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast6 active
```

To remove multicast traffic reduction configurations in VLAN 2, and take the global multicast traffic reduction configuration, enter the following command.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# no multicast6 active
```

Syntax: [**no**] **multicast6 active** | **passive**

When you enable IPv6 multicast for a specific VLAN, MLD snooping is enabled. The device uses MLD to maintain a table of the Group Membership reports received by the device for the specified VLAN. You can use active or passive MLD mode. There is no default mode.

The description for the MLD modes is as follows:

- **Active** – When active MLD mode is enabled, the router actively sends out MLD queries to identify IPv6 multicast groups within the VLAN and makes entries in the MLD table based on the Group Membership reports received from the network.
- **Passive** – When passive MLD mode is enabled, the router listens for MLD Group Membership reports on the VLAN specified but does not send MLD queries. The passive mode is called "MLD snooping". Use this mode when another device in the VLAN is actively sending queries.

Configuring MLD proxy per VLAN

The **multicast6 mld-proxy-enable** command enables MLD proxy for IPv6. Using the MLD proxy function, the host is able send out MLD reports on behalf of the hosts behind the switch.

To configure a device to function as an MLD proxy on VLAN 2, enter the following commands as shown in this example.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast6 active
NetIron(config-vlan-2)# multicast6 mld-proxy-enable
```

Syntax: [**no**] **multicast6 mld-proxy-enable**

The **no** form of this command disables MLD proxy on a per-VLAN basis.

Configuring the PIM-SM traffic snooping per VLAN

In the following example, multicast traffic reduction is applied using PIM-SM traffic snooping to VLAN 2.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast6 active
NetIron(config-vlan-2)# multicast6 pimsm-snooping
```

Syntax: [**no**] **multicast6 pimsm-snooping**

The **no** form of this command disables PIM SM traffic snooping on a per-VLAN basis.

Configuring PIM proxy per VLAN

Using the PIM proxy function, multicast traffic can be reduced by configuring a device to issue PIM join and prune messages on behalf of hosts that the configured router discovers through standard PIM interfaces. The router is then able to act as a proxy for the discovered hosts and perform PIM tasks upstream of the discovered hosts. Where there are multiple PIM downstream routers, this removes the need to send multiple messages.

When configuring PIM proxy on a VLAN, you must first configure PIM-SM traffic snooping. To configure a device to function as a PIM proxy on VLAN 2, use the following commands.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast6 active
NetIron(config-vlan-2)# multicast6 pimsm-snooping
NetIron(config-vlan-2)# multicast6 pim-proxy-enable
```

Syntax: [**no**] **multicast6 pim-proxy-enable**

The **no** form of this command disables PIM proxy on a per-VLAN basis.

Configuring an IPv6 multicast static group uplink per VLAN

When the **multicast6 static-group uplink** command is enabled on a snooping VLAN, the snooping device behaves like an MLD host on ports connected to the multicast router. The snooping device will respond to MLD queries from the uplink multicast PIM router for the groups and sources configured. Upon the multicast router receiving the MLD join message, it will initiate the PIM join on its upstream path towards the source to pull the source traffic down. The source traffic will stop at the MLD snooping device. The traffic will then be forwarded to the multicast receiver and router ports or dropped in hardware if no other multicast receiver and routers are present in the VLAN.

The **multicast6 static-group uplink** command can be configured under the VLAN configuration only.

The **multicast6 static-group uplink** command must be used with the **multicast6 static-group** command in order to connect a remote multicast source with the snooping VLAN where the static group is configured.

When using MLDv2, you can use the **multicast6 static-group include** or **multicast6 static-group exclude** command to statically *include* or *exclude* multicast traffic, respectively for hosts that cannot signal group membership dynamically.

To configure the snooping device to statically join a multicast group on the uplink interface, enter the following commands.

```
NetIron(config)# vlan 10
```

```
NetIron(config-vlan-10)# multicast6 static-group active
NetIron(config-vlan-10)# multicast6 static-group ff2e::1 uplink
```

NOTE
The following error message will display if both static uplink MLDv1 and MLDv2 are configured for the same group: Error: Static v1 and v2 uplink configuration cannot co-exist for the same group.

To configure the physical interface Ethernet 1/1 to statically join a multicast group with an IPv6 group address of ff0e::1, enter commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# multicast6 static-group ff0e::1 ethernet 1/1
```

To configure the snooping device to statically join a multicast stream on the uplink interface with the source address of 2003::1 in the MLDv2 include mode, enter commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# multicast6 static-group ff1e::1 include 2003::1 uplink
```

To configure the snooping device to statically join all multicast streams on the uplink interface excluding the stream with source address 2002::1, enter commands such as the following.

```
NetIron(config)# vlan 10
NetIron(config-vlan-10)# multicast6 static-group active
NetIron(config-vlan-10)# multicast6 static-group ff1e::1 exclude 2002::1 uplink
```

Syntax: [**no**] **multicast6 static-group** <*group-address*> **uplink**

Syntax: [**no**] **multicast6 static-group** <*group-address*> [**include** | **exclude** <*source-address*>] **uplink**

The <*group-address*> variable specifies the group IPv6 multicast address.

The **include** or **exclude** keyword indicates a filtering action. You can specify which source (for a group) to include or exclude. The **include** or **exclude** keyword is only supported on MLDv2.

The <*source-address*> parameter specifies the IPv6 address of the multicast source. Each address must be added or deleted one line per source.

The **uplink** parameter specifies the port as an uplink port that can receive multicast data for the configured multicast groups. Upstream traffic will be sent to the router and will not use a port.

The **no** form of this command removes the static multicast definition. Each configuration must be deleted separately.

Configuring IPv6 multicast tracking and fast-leave per VLAN

The **multicast6 tracking** command enables IPv6 multicast tracking per VLAN. The **multicast6 tracking** command is similar to the **ipv6 multicast tracking** command. When the **multicast6 tracking** command is configured, MLD fast-leave is enabled. For more information on MLD fast-leave for IPv6 multicast tracking, refer to "Configuring IPv6 multicast tracking and fast-leave" on page 1998.

To enable IPv6 multicast tracking per VLAN, enter commands such as the following.

```
NetIron(config)# vlan 2
NetIron(config-vlan-2)# multicast6 active
NetIron(config-vlan-2)# multicast6 tracking
```

Syntax: [**no**] **multicast6 tracking**

The **no** form of this command disables the tracking process.

Displaying IPv6 multicast information

To display information for IPv6 multicast traffic reduction configuration, enter the following command.

```
NetIron(config)# show ipv6 multicast
Global Multicast Traffic Reduction Configuration
   MLD Snooping State :   Disabled    Version              :          1
   Group Interval     :        260    Query Interval       :        125
   Max Response Time  :         10    Robustness Var       :          1
   Last Member Qry Int:          5    Last Member Qry Count:          3
   Querier Exp Tm     :        255
   MLD Proxy          :   Disabled    Proxy Interval       :         60
   Filter             :   Disabled    Tracking             :   Disabled

   PIM Snooping       :   Disabled
   PIM Prune Wait Time:          3
   PIM Proxy          :   Disabled    Proxy Interval       :         60
VLAN snooping configurations:

VLAN ID 2
 IPv6 Multicast snooping is enabled - Active. Entries 0
 IPv6 Multicast MLD tracking is disabled
VLAN ID 10
 IPv6 Multicast snooping is enabled - Active. Entries 0
IPv6 Multicast pimsm snooping is enabled
```

Syntax: **show ipv6 multicast** [**mldv2** <*vlan-id*> | **pim** <*vlan-id*> | **static** <*vlan-id*> | **statistics** <*vlan-id*> | **tracking** <*vlan-id*> | **vlan** <*vlan-id*>]

The **mldv2** <*vlan-id*> parameter displays IPv6 multicast information specific to MLDv2 for a specified port-based VLAN.

The **pim** <*vlan-id*> parameter displays IPv6 multicast information specific to PIM for a specified port-based VLAN.

The **static** <*vlan-id*> parameter displays information for the number of static MLD snooping entries configured in a specified port-based VLAN.

The **statistics** <*vlan-id*> parameter displays IPv6 multicast statistics for a specified port-based VLAN.

The **tracking** <*vlan-id*> parameter displays tracking information for MLDv2 hosts for a specified port-based VLAN.

The **vlan** <*vlan-id*> parameter displays IPv6 multicast PIM information for a specified port-based VLAN. The <*vlan-id*> variable is entered in decimal format.

Table 386 describes the fields displayed by the **show ipv6 multicast** command.

TABLE 386 Output from the **show ipv6 multicast** command

Field	Description
Global Multicast Traffic Reduction Configuration	Indicates all multicast traffic configuration displayed for all parameters.
MLD Snooping State:	Indicates whether MLD snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Global Interval	Indicates the time until the groups time out if no reports are received.

TABLE 386 Output from the **show ipv6 multicast** command (Continued)

Field	Description
Max Response Time	The length of time in seconds that the router will wait for an IGMP (V1 or V2) response from an interface before concluding that the group member on that interface is down and removing it from the group.
Last Member Qry Int	Indicates when a leave is received; a group-specific query is sent. The last member query count is the number of queries with a time interval of (LMQT) is sent.
Querier Exp Tm	Indicates the time until the querier times out if no query is received.
MLD Proxy	Indicates whether MLD Proxy is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Filter	Indicates whether filtering is enabled or disabled. Filtering multicast groups allows the device to filter out all multicast traffic groups other than the ones for which the device has received Group Membership reports.
PIM Snooping	Indicates if PIM snooping is enabled. If disabled, this line does not appear.
PIM Prune Wait Time	The amount of time a PIM router will wait before stopping traffic to neighbor routers that do not want the traffic. The value can be from 0 to 3 seconds. The default is 3 seconds.
PIM Proxy	Indicates whether PIM proxy is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Version	The MLD version (1 or 2) operating on the router.
Query Interval	How often the router will query an interface for group membership.
Robustness Var	Used to fine tune for unexpected loss on the subnet. The value is used to calculate the group interval.
Last Member Qry Count	Specifies the number of group-specific queries when a leave is received.
Proxy Interval	Indicates the time interval in seconds between PIM proxy and join messages. You can specify a value from 10 through 600 seconds.
Tracking	Indicates whether tracking is enabled or disabled.
VLAN ID	The port-based VLAN to which the information listed below the ID applies.
IPv6 multicast traffic snooping	Indicates whether IPv6 multicast traffic snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
IPv6 Multicast MLD tracking	Indicates whether IPv6 multicast MLD tracking is enabled or disabled.
IPV6 Multicast pimsm snooping	Indicates whether pimsm snooping is enabled or disabled.

To display detailed IPv6 multicast traffic reduction information for a specified VLAN, enter the following command at any level of the CLI.

```
NetIron# show ipv6 multicast vlan 60
L2mdb port type: R-router port, V1-mld v1, V2-mld v2, P_SG-pim sg, P_G-pim g
VLAN ID 60
IPv6 Multicast snooping is running - Passive
IPv6 Multicast mld operating version  - 1 (7s)
 Router ports: 4/2 (9s)
 Number of Multicast Groups: 1

  1    Group: ff09::9
       Ports: 1/1 vlan 60 type  MLDv1  (7s)
              4/2 vlan 60 type  R (9s)
       1    Source: (2060::12,  4/2) FID 0x800a mvid none
            SG group ports: 1/1(1/1) vlan 60 type  MLDv1  (0s)
                            1/2(1/2) vlan 60 type  MLDv1  (0s)
```

Syntax: show ipv6 multicast [**vlan** <*vlan-id*>]

The **vlan** <*vlan-id*> parameter displays IPv6 multicast PIM information for a specified port-based VLAN. The <*vlan-id*> variable is entered in decimal format.

Table 387 displays the output from the **show ipv6 multicast vlan** command.

TABLE 387 Output from the **show ipv6 multicast** command

Field	Description
L2mdb port type	Specifies if it is a router port, or an S,G port, or G port.
VLAN ID	The VLAN to which the information listed below the ID applies. The VLAN ID displays IPv6 multicast PIM information.
IPv6 multicast snooping is running	Indicates whether IPv6 multicast snooping is running as enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
IPv6 Multicast mld operating version	Indicates whether IPv6 Multicast MLD version is 1 or 2.
Router Ports	The ports that are connected to routers that support IP multicast.
Number of Multicast Groups	The total number of groups for which the VLAN ports have received MLD group membership reports, join messages, or prune messages.
Group	An IPV6 multicast group.
Ports	Indicates outgoing ports of the group.
Source	The IPv6 address of each PIM SM source, and the device ports that are connected to the receivers of the source.
SG group ports	Indicates outgoing ports for an S,G entry.

Chapter

Managing a Device Over IPv6

47

The following ways to Manage a Device Over IPv6 features are supported by PowerConnect B-MLXe .Series devices.

- IPv6 copy Command
- Copying a File from an IPv6 TFTP Server
- Using the IPv6 ncopy Command
- IPv6 Ping Command
- IPv6 Traceroute Command
- IPv6 Telnet
- Secure Shell

You can perform system management tasks for the device using the **copy**, **ncopy**, **ping**, **telnet**, and **traceroute** commands and Secure Shell (SSH). These commands and SSH now function over IPv6.

This section describes the IPv6-related syntax added to the commands and SSH. It does not describe the already existing command syntax for IPv4.

Using the IPv6 copy command

The **copy** command for IPv6 allows you to do the following:

- Copy a file from a specified source to an IPv6 TFTP server.
- Copy a file from an IPv6 TFTP server to a specified destination.

Copying a file to an IPv6 TFTP server

You can copy a file from the following sources to an IPv6 TFTP server:

- Flash memory.
- Running configuration.
- Startup configuration.

Copying a file from flash memory

For example, to copy the primary or secondary boot image from the device's flash memory to an IPv6 TFTP server, enter a command such as the following.

```
NetIron# copy flash tftp 2001:7382:e0ff:7837::3 test.img secondary
```

This command copies the secondary boot image named test.img from flash memory to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3.

Syntax: copy flash tftp <*ipv6-address*> <*source-file-name*> **primary | secondary**

The <*ipv6-address*> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*source-file-name*> parameter specifies the name of the file you want to copy to the IPv6 TFTP server.

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

Copying a file from the running or startup configuration

For example, to copy the running configuration to an IPv6 TFTP server, enter a command such as the following.

```
NetIron# copy running-config tftp 2001:7382:e0ff:7837::3 newrun.cfg
```

This command copies the running configuration to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 and names the file on the TFTP server newrun.cfg.

Syntax: copy running-config | startup-config tftp <*ipv6-address*> <*destination-file-name*>

Specify the **running-config** keyword to copy the running configuration file to the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration file to the specified IPv6 TFTP server.

The tftp <*ipv6-address*> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*destination-file-name*> parameter specifies the name of the file that is copied to the IPv6 TFTP server.

Copying a file from an IPv6 TFTP server

You can copy a file from an IPv6 TFTP server to the following destinations:

- Flash memory.
- Running configuration.
- Startup configuration.

Copying a file to flash memory

For example, to copy a boot image from an IPv6 TFTP server to the primary or secondary storage location in the device's flash memory, enter a command such as the following.

```
NetIron# copy tftp flash 2001:7382:e0ff:7837::3 test.img secondary
```

This command copies a boot image named test.img from an IPv6 TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the secondary storage location in the device's flash memory.

Syntax: copy tftp flash <*ipv6-address*> <*source-file-name*> **primary | secondary**

The <*ipv6-address*> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*source-file-name*> parameter specifies the name of the file you want to copy from the IPv6 TFTP server.

The **primary** keyword specifies the primary storage location in the device's flash memory, while the **secondary** keyword specifies the secondary storage location in the device's flash memory.

Copying a file to the running or startup configuration

For example, to copy a configuration file from an IPv6 TFTP server to the router's running or startup configuration, enter a command such as the following.

```
NetIron# copy tftp running-config 2001:7382:e0ff:7837::3 newrun.cfg overwrite
```

This command copies the newrun.cfg file from the IPv6 TFTP server and overwrites the router's running configuration file with the contents of newrun.cfg.

NOTE
To activate this configuration, you must reload (reset) the device.

Syntax: copy tftp running-config | startup-config <*ipv6-address*> <*source-file-name*> **[overwrite]**

Specify the **running-config** keyword to copy the running configuration from the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration from the specified IPv6 TFTP server.

The <*ipv6-address*> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*source-file-name*> parameter specifies the name of the file that is copied from the IPv6 TFTP server.

The **overwrite** keyword specifies that the device should overwrite the current configuration file with the copied file. If you do not specify this parameter, the device copies the file into the current running or startup configuration but does not overwrite the current configuration.

NOTE
You cannot use the overwrite option from non-console sessions, because it will disconnect the session.

Using the IPv6 ncopy command

The **ncopy** command for IPv6 allows you to do the following:

- Copy a primary or secondary boot image from flash memory to an IPv6 TFTP server.
- Copy the running configuration to an IPv6 TFTP server.
- Copy the startup configuration to an IPv6 TFTP server
- Upload various files from an IPv6 TFTP server.

Copying a primary or secondary boot image from flash memory to an IPv6 TFTP server

For example, to copy the primary or secondary boot image from the device's flash memory to an IPv6 TFTP server, enter a command such as the following.

```
NetIron# ncopy flash primary tftp 2001:7382:e0ff:7837::3 primary.img
```

This command copies the primary boot image named primary.img from flash memory to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3.

Syntax: ncopy flash primary | secondary tftp <*ipv6-address*> <*source-file-name*>

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

The **tftp** <*ipv6-address*> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*source-file-name*> parameter specifies the name of the file you want to copy from flash memory.

Copying the running or startup configuration to an IPv6 TFTP server

For example, to copy a device's running or startup configuration to an IPv6 TFTP server, enter a command such as the following.

```
NetIron# ncopy running-config tftp 2001:7382:e0ff:7837::3 bakrun.cfg
```

This command copies a device's running configuration to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 and names the destination file bakrun.cfg.

Syntax: ncopy running-config | startup-config tftp <*ipv6-address*> <*destination-file-name*>

Specify the **running-config** keyword to copy the device's running configuration or the **startup-config** keyword to copy the device's startup configuration.

The **tftp** <*ipv6-address*> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*destination-file-name*> parameter specifies the name of the running configuration that is copied to the IPv6 TFTP server.

Uploading files from an IPv6 TFTP server

You can upload the following files from an IPv6 TFTP server:

- Primary boot image.
- Secondary boot image.
- Running configuration.
- Startup configuration.

Uploading a primary or secondary boot image from an IPv6 TFTP server

For example, to upload a primary or secondary boot image from an IPv6 TFTP server to a device's flash memory, enter a command such as the following.

```
NetIron# ncopy tftp 2001:7382:e0ff:7837::3 primary.img flash primary
```

This command uploads the primary boot image named primary.img from a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the device's primary storage location in flash memory.

Syntax: ncopy tftp <*ipv6-address*> <*source-file-name*> **flash primary | secondary**

The **tftp** <*ipv6-address*> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*source-file-name*> parameter specifies the name of the file you want to copy from the TFTP server.

The **primary** keyword specifies the primary location in flash memory, while the **secondary** keyword specifies the secondary location in flash memory.

Uploading a running or startup configuration from an IPv6 TFTP server

For example to upload a running or startup configuration from an IPv6 TFTP server to a device, enter a command such as the following.

```
NetIron# ncopy tftp 2001:7382:e0ff:7837::3 newrun.cfg running-config
```

This command uploads a file named newrun.cfg from a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the device.

Syntax: ncopy tftp <*ipv6-address*> <*source-file-name*> **running-config | startup-config**

The **tftp** <*ipv6-address*> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*source-file-name*> parameter specifies the name of the file you want to copy from the TFTP server.

Specify the **running-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The device copies the specified file into the current running configuration but does not overwrite the current configuration.

Specify the **startup-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The device copies the specified file into the current startup configuration but does not overwrite the current configuration.

Using the IPv6 ping command

The **ping** command allows you to verify the connectivity from a device to an IPv6 device by performing an ICMP for IPv6 echo test.

For example, to ping a device with the IPv6 address of 2001:3424:847f:a385:34dd::45 from the device, enter the following command.

```
NetIron# ping ipv6 2001:3424:847f:a385:34dd::45
```

Syntax: **ping ipv6** <*ipv6-address*> [**outgoing-interface** [<*port*> | **ve** <*number*>]] [**source** <*ipv6-address*>] [**count** <*number*>] [**timeout** <*milliseconds*>] [**ttl** <*number*>] [**size** <*bytes*>] [**quiet**] [**numeric**] [**no-fragment**] [**verify**] [**data <1-to-4 byte hex>**] [**brief**]

The <*ipv6-address*> parameter specifies the address of the router. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **outgoing-interface** keyword specifies a physical interface over which you can verify connectivity. If you specify a physical interface, such as an Ethernet interface, you must also specify the port number of the interface. If you specify a virtual interface, such as a VE, you must specify the number associated with the VE.

The **source** <*ipv6-address*> parameter specifies an IPv6 address to be used as the origin of the ping packets.

The **count** <*number*> parameter specifies how many ping packets the router sends. You can specify from 1 - 4294967296. The default is 1.

The **timeout** <*milliseconds*> parameter specifies how many milliseconds the router waits for a reply from the pinged device. You can specify a timeout from 1 - 4294967296 milliseconds. The default is 5000 (5 seconds).

The **ttl** <*number*> parameter specifies the maximum number of hops. You can specify a TTL from 1 - 255. The default is 64.

The **size** <*bytes*> parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 - 4000. The default is 16.

The **no-fragment** keyword turns on the "don't fragment" bit in the IPv6 header of the ping packet. This option is disabled by default.

The **quiet** keyword hides informational messages such as a summary of the ping parameters sent to the device and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.

The **verify** keyword verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.

The **data** <*1 - 4 byte hex*> parameter lets you specify a specific data pattern for the payload instead of the default data pattern, "abcd", in the packet's data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE
For parameters that require a numeric value, the CLI does not check that the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

The **brief** keyword causes ping test characters to be displayed. The following ping test characters are supported:

! Indicates that a reply was received.

. Indicates that the network server timed out while waiting for a reply.

U Indicates that a destination unreachable error PDU was received.

I Indicates that the user interrupted ping.

Using the IPv6 traceroute command

The **traceroute** command allows you to trace a path from the device to an IPv6 host.

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a minimum TTL of 1 second and a maximum TTL of 30 seconds. In addition, if there are multiple equal-cost routes to the destination, the device displays up to three responses.

For example, to trace the path from the device to a host with an IPv6 address of 3301:23dd:349e:a384::34, enter the following command.

```
NetIron# traceroute ipv6 3301:23dd:349e:a384::34
```

Syntax: traceroute ipv6 <*ipv6-address*>

The <*ipv6-address*> parameter specifies the address of a host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Using Telnet

This section explains how to do the following:

- Use the **telnet** command to establish a Telnet session from the device to a remote IPv6 host.
- Establish a Telnet session from a remote IPv6 host to the device.

Using the IPv6 Telnet command

The **telnet** command allows a Telnet connection from a device to a remote IPv6 host using the console. Up to five read-access and one write-access inbound Telnet session are supported on the router at one time. Up to five simultaneous outbound Telnet sessions can also be supported from the console session, from inbound Telnet sessions, from inbound SSH sessions or from a Web session.

To see the Telnet sessions currently open on the device, enter the **show telnet** command; to see both the open Telnet and open SSH sessions, enter the **show who** command as shown below.

```
NetIron# show who
Console connections:
       established
       3 days 17 hours 31 minutes 27 seconds in idle
Telnet server status: Enabled
Telnet connections (inbound):
 1      established, client ip address 10.53.1.86
       you are connecting to this session
       1 seconds in idle
 2      established, client ip address 10.53.1.86
       7 seconds in idle
 3      closed
 4      closed
 5      closed
Telnet connections (outbound):
 6      established, server ip address 10.47.2.200, from Telnet session 1
       4 seconds in idle
 7      closed
 8      closed
 9      closed
 10     closed
SSH server status: Enabled
SSH connections:
 1      closed
 2      closed
 3      closed
 4      closed
...
```

Syntax: show who

To establish a Telnet connection to a remote host, use the **telnet** command. The following example will establish an outbound Telnet connection to a remote host with the IPv6 address of 3001:2837:3de2:c37::6.

```
NetIron# telnet 3001:2837:3de2:c37::6
```

Syntax: telnet <*ipv6-address*> [<*port-number*> **| outgoing-interface ethernet** <*port*> **| ve** <*number*>]

The <*ipv6-address*> parameter specifies the address of a remote host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <*port-number*> parameter specifies the port number on which the device establishes the Telnet connection. You can specify a value between 1 - 65535. If you do not specify a port number, the device establishes the Telnet connection on port 23.

If the IPv6 address you specify for the **telnet** <*ipv6-address*> command is a link-local address, you must specify the **outgoing-interface** ethernet <*port*> | ve <*number*> parameter. This parameter specifies the interface that must be used to reach the remote host. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

Establishing a Telnet session from an IPv6 host

To establish a Telnet session from an IPv6 host to the device, open your Telnet application and specify the IPv6 address of the router.

Using Secure Shell

Secure Shell (SSH) is a mechanism that allows secure remote access to management functions on the device. SSH provides a function similar to Telnet. You can log into and configure the device using a publicly or commercially available SSH client program, just as you can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the device.

To open an SSH session from an IPv6 host running an SSH client program to the device, open your SSH client program and specify the IPv6 address of the router.

Chapter

Configuring Secure Shell and Secure Copy 48

The following Secure Shell features are supported by PowerConnect B-MLXe Series devices.

- SSH Transport Layer Protocol
- SSH Authentication Protocol
- SSH Connection Protocol
- SSH Fingerprint Format
- SSH Protocol Assigned Numbers
- SCP for Copying code images
- SCP for Copying code images
- SSH Transport Layer Encryption Modes
- CP or SFTP or SSH URI Format
- DSA challenge-response authentication
- Password authentication
- 3DES as the encryption algorithm
- AES as the encryption algorithm
- SHA 1 as the MAC algorithm

Secure Shell (SSH) is a mechanism for allowing secure remote access to management functions on a PowerConnect. SSH provides a function similar to Telnet. Users can log into and configure the device using a publicly or commercially available SSH client program, just as they can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the device.

SSH v2 is supported on the PowerConnect. Dell's SSHv2 implementation is compatible with all versions of the SSHv2 protocol (2.1, 2.2, and so on). At the beginning of an SSH session, the PowerConnect negotiates the version of SSHv2 to be used. The highest version of SSHv2 supported by both the PowerConnect and the client is the version that is used for the session. Once the SSHv2 version is negotiated, the encryption algorithm with the highest security ranking is selected to be used for the session.

The number of SSH client instances allowed is16.

Also, the PowerConnect support Secure Copy (SCP) for securely transferring files between a PowerConnect and an SCP-enabled remote hosts. Refer to "Using Secure Copy" on page 2026 for more information.

NOTE
SSH is disabled by default. To gain access to a PowerConnect router through SSH, you must enable it as described in this chapter.

SSH Version 2 support

SSHv2 is a substantial revision of Secure Shell, comprising the following hybrid protocols and definitions:

- SSH Transport Layer Protocol
- SSH Authentication Protocol
- SSH Connection Protocol
- SECSH Public Key File Format
- SSH Fingerprint Format
- SSH Protocol Assigned Numbers
- SSH Transport Layer Encryption Modes
- SCP or SFTP or SSH URI Format

If you are using redundant management modules, you can synchronize the DSA host key pair between the active and standby modules by entering the **sync-standby** command at the Privileged EXEC level of the CLI.

Tested SSHv2 clients

The following SSH clients have been tested with SSHv2:

- SSH Secure Shell 3.2.3
- Van Dyke SecureCRT 4.0 and 4.1
- F-Secure SSH Client 5.3 and 6.0
- PuTTY 0.54 and 0.56

NOTE
On the PuTTy client, under the options that control key re-exchange, it is recommended that the maximum minutes before rekey be set to 0 and the maximum data before rekey be set to 0.

- OpenSSH 3.5_p1 and 3.6.1p2
- Solaris Sun-SSH-1.0

Supported features

The SSH server allows secure remote access management functions on a device. SSH provides a function that is similar to Telnet, but unlike Telnet, SSH provides a secure, encrypted connection.

SSHv2 support includes the following:

- The following encryption cipher algorithm are supported. They are listed in order of preference:
  - **aes256-cbc:** AES in CBC mode with 256-bit key
  - **aes192-cbc:** AES in CBC mode with 192-bit key
  - **aes128-cbc:** AES in CBC mode with 128-bit key
  - **3des-cbc:** Triple-DES
- Key exchange methods, in the order of preference are:
  - **diffie-hellman-group1-sha1**

  - **diffie-hellman-group14-sha1**
- Public key algorithm is **ssh-dss**.
- Data integrity is ensured with **hmac-sha1** algorithm.
- Supported authentication methods are **Password** and **publickey**.
- Compression is not supported.
- TCP or IP port forwarding, X11 forwarding, and secure file transfer are not supported.
- SSH version 1 is not supported.
- SCP supports AES encryption

Configuring SSH

Dell's implementation of SSH supports two kinds of user authentication:

- ***DSA challenge-response authentication***, where a collection of public keys are stored on the device. Only clients with a private key that corresponds to one of the stored public keys can gain access to the device using SSH.
- ***Password authentication***, where users attempting to gain access to the device using an SSH client are authenticated with passwords stored on the device or on a TACACS or TACACS+ or RADIUS server

Both kinds of user authentication are enabled by default. You can configure the device to use one or both of them.

To configure Secure Shell on a PowerConnect, perform the tasks listed below.

1. Generate a host DSA public and private key pair for the device.
2. Configure DSA challenge-response authentication.
3. Set optional parameters.

You can also view information about active SSH connections on the device as well as terminate them.

Generating a host key pair

When SSH is configured, a public and private ***host DSA key pair*** is generated for the PowerConnect. The SSH server on the PowerConnect uses this host DSA key pair, along with a dynamically generated ***server DSA key pair***, to negotiate a session key and encryption method with the client trying to connect to it.

The host DSA key pair is stored in the PowerConnect's system-config file. Only the public key is readable. The public key should be added to a "known hosts" file (for example, $HOME/.ssh/known_hosts on UNIX systems) on the clients who want to access the device. Some SSH client programs add the public key to the known hosts file automatically; in other cases, you must manually create a known hosts file and place the PowerConnect's public key in it. Refer to "Providing the public key to clients" on page 2018 for an example of what to place in the known hosts file.

While the SSH listener exists at all times, sessions can't be started from clients until a key is generated. Once a key is generated, clients can start sessions. The keys are also not displayed in the configuration file by default. To display the keys, use the **ssh show-host-keys** command in Privileged EXEC mode. To generate a public and private DSA host key pair on a PowerConnect, enter the following commands.

```
NetIron(config)# crypto key generate
```

When a host key pair is generated, it is saved to the flash memory of all management modules.

To disable SSH in SSHv2 on a PowerConnect, enter the following commands.

```
NetIron(config)# crypto key zeroize
```

When SSH is disabled, it is deleted from the flash memory of all management modules.

Syntax: crypto key generate | zeroize

The **generate** keyword places an DSA host key pair in the flash memory and enables SSH on the device.

The **zeroize** keyword deletes the DSA host key pair from the flash memory and disables SSH on the device.

By default, public keys are hidden in the running configuration. You can optionally configure the PowerConnect to display the DSA host key pair in the running configuration file entering the following command.

```
NetIron# ssh show-host-keys
```

Syntax: ssh show-host-keys

To hide the public keys in the running configuration file, enter the following command.

```
NetIron# ssh no-show-host-keys
```

Syntax: ssh no-show-host-keys

Providing the public key to clients

If you are using SSH to connect to a PowerConnect from a UNIX system, you may need to add the PowerConnect's public key to a "known hosts" file; for example, $HOME/.ssh/known_hosts. The following is an example of an entry in a known hosts file.

```
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
```

Configuring DSA challenge-response authentication

With DSA challenge-response authentication, a collection of clients' public keys are stored on the PowerConnect. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

When DSA challenge-response authentication is enabled, the following events occur when a client attempts to gain access to the device using SSH.

1. The client sends its public key to the PowerConnect.
2. The PowerConnect compares the client's public key to those stored in memory.
3. If there is a match, the PowerConnect uses the public key to encrypt a random sequence of bytes.
4. The PowerConnect sends these encrypted bytes to the client.
5. The client uses its private key to decrypt the bytes.
6. The client sends the decrypted bytes back to the PowerConnect.
7. The PowerConnect compares the decrypted bytes to the original bytes it sent to the client. If the two sets of bytes match, it means that the client's private key corresponds to an authorized public key, and the client is authenticated.

Setting up DSA challenge-response authentication consists of the following steps.

1. Importing authorized public keys into the PowerConnect.
2. Enabling DSA challenge response authentication

Importing authorized public keys into the PowerConnect

SSH clients that support DSA authentication normally provide a utility to generate an DSA key pair. The private key is usually stored in a password-protected file on the local host; the public key is stored in another file and is not protected. You should collect one public key from each client to be granted access to the PowerConnect and place all of these keys into one file. This public key file is imported into the PowerConnect.

The following is an example of a public key file containing one public keys.

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

NOTE
Make sure the key ends with the complete phrase "`---- END SSH2 PUBLIC KEY ----`" before importing the public key. Otherwise, a warning is displayed whenever the device is reloaded.

You can import the authorized public keys into the active configuration by loading them from a file on a TFTP server and are saved on the EEPROM of the chassis. If you import a public key file from a TFTP server, the file is automatically loaded into the active configuration the next time the device is booted.

To cause a public key file called pkeys.txt to be loaded from a TFTP server each time the PowerConnect is booted, enter a command such as the following.

```
NetIron(config)# ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
```

Syntax: ip ssh pub-key-file tftp ipv6 <*ipv6-addr*> **|** <*tftp-server-ip-addr*> <*filename*> **[remove]**

The <*tftp-server-ip-addr*> variable is the IP address of the tftp server that contains the public key file that you want to import into the device.

The <*filename*> variable is the name of the dsa public key file that you want to import into the device.

The **remove** parameter deletes the key from the system.

To display the currently loaded public keys, enter the following command.

```
NetIron# show ip client-pub-key
---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

Syntax: show ip client-pub-key [| begin<*expression*> **| exclude** <*expression*> **| include** <*expression*>**]**

To clear the public keys from the buffers, enter the following command.

```
NetIron# clear public-key
```

Syntax: clear public-key

Use the **ip ssh pub-key remove** command to delete the public key from the system.

Enabling DSA challenge-response authentication

DSA challenge-response authentication is enabled by default. You can disable or re-enable it manually.

To enable DSA challenge-response authentication.

```
NetIron(config)# ip ssh key-authentication yes
```

To disable DSA challenge-response authentication.

```
NetIron(config)# ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

Setting optional parameters

You can adjust the following SSH settings on the device:

- Number of SSH authentication retries
- User authentication method the device uses for SSH connections
- Whether or not the device allows users to log in without supplying a password

- Port number for SSH connections
- SSH login timeout value
- A specific interface to be used as the source for all SSH traffic from the device
- Maximum idle time for SSH sessions
- Disable 3-DES support

Setting the number of SSH authentication retries

By default, the PowerConnect attempts to negotiate a connection with the connecting host three times. The number of authentication retries can be changed to between 1 – 5.

For example, the following command changes the number of authentication retries to 5.

```
NetIron(config)# ip ssh authentication-retries 5
```

Syntax: ip ssh authentication-retries <*number*>

Deactivating user authentication

After the SSH server on the PowerConnect negotiates a session key and encryption method with the connecting client, user authentication takes place. Dell's implementation of SSH supports DSA challenge-response authentication and password authentication.

With DSA challenge-response authentication, a collection of clients' public keys are stored on the PowerConnect. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

With password authentication, users are prompted for a password when they attempt to log into the device (provided empty password logins are not allowed; refer to "Enabling empty password logins" on page 2022). If there is no user account that matches the user name and password supplied by the user, the user is not granted access.

You can deactivate one or both user authentication methods for SSH. Note that deactivating both authentication methods essentially disables the SSH server entirely.

To disable DSA challenge-response authentication.

```
NetIron(config)# ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

The default is "yes".

To deactivate password authentication.

```
NetIron(config)# ip ssh password-authentication no
```

Syntax: ip ssh password-authentication no | yes

The default is "yes".

Enabling empty password logins

By default, empty password logins are not allowed. This means that users with an SSH client are always prompted for a password when they log into the device. To gain access to the device, each user must have a user name and password. Without a user name and password, a user is not granted access. Refer to "Setting up local user accounts" on page 31 for information on setting up user names and passwords on the PowerConnect.

If you enable empty password logins, users are ***not*** prompted for a password when they log in. Any user with an SSH client can log in without being prompted for a password.

To enable empty password logins.

```
NetIron(config)# ip ssh permit-empty-passwd yes
```

Syntax: ip ssh permit-empty-passwd no | yes

Setting the SSH port number

By default, SSH traffic occurs on TCP port 22. You can change this port number. For example, the following command changes the SSH port number to 2200.

```
NetIron(config)# ip ssh port 2200
```

Note that if you change the default SSH port number, you must configure SSH clients to connect to the new port. Also, you should be careful not to assign SSH to a port that is used by another service. If you change the SSH port number, it is recommended that you change it to a port number greater than 1024.

Syntax: ip ssh port <*number*>

Setting the SSH login timeout value

When the SSH server attempts to negotiate a session key and encryption method with a connecting client, it waits a maximum of 120 seconds for a response from the client. If there is no response from the client after 120 seconds, the SSH server disconnects. You can change this timeout value to between 1 – 120 seconds. For example, to change the timeout value to 60 seconds.

```
NetIron(config)# ip ssh timeout 60
```

Syntax: ip ssh timeout <*seconds*>

NOTE
The standard for the idle-timeout RADIUS attribute is for it to be implemented in seconds as opposed to the minutes that the PowerConnect router uses. If this attribute is used for setting idle time instead of this configuration, the value from the idle-timeout RADIUS attribute will be converted to seconds and truncated to the nearest minute. Designating an Interface as the Source for All SSH Packets

---

You can designate a loopback interface, virtual interface, or Ethernet port as the source for all SSH packets from the device. The software uses the IP address with the numerically lowest value configured on the port or interface as the source IP address for SSH packets originated by the device.

NOTE
When you specify a single SSH source, you can use only that source address to establish SSH management sessions with the PowerConnect.

To specify the numerically lowest IP address configured on a loopback interface as the device's source for all SSH packets, enter commands such as a the following.

```
NetIron(config)# int loopback 2
NetIron(config-lbif-2)# ip address 10.0.0.2/24
NetIron(config-lbif-2)# exit
NetIron(config)# ip ssh source-interface loopback 2
```

The commands in this example configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all SSH packets from the PowerConnect.

Syntax: ip ssh source-interface ethernet <*slot/port*> **| loopback** <*num*> **| ve** <*num*>

The <*num*> parameter is a loopback interface or virtual interface number. The <*slot/port*> parameter specifies an ethernet port number.

Example

```
NetIron(config)# interface ethernet 1/4
NetIron(config-if-e10000-1/4)# ip address 209.157.22.110/24
NetIron(config-if-e10000-1/4)# exit
NetIron(config)# ip ssh source-interface ethernet 1/4
```

Configuring maximum idle time for SSH sessions

By default, SSH sessions do not time out. Optionally, you can set the amount of time an SSH session can be inactive before the PowerConnect closes it. For example, to set the maximum idle time for SSH sessions to 30 minutes.

```
NetIron(config)# ip ssh idle-time 30
```

Syntax: ip ssh idle-time <*minutes*>

If an established SSH session has no activity for the specified number of minutes, the PowerConnect closes it. An idle time of 0 minutes (the default value) means that SSH sessions never time out. The maximum idle time for SSH sessions is 240 minutes.

NOTE
The standard for the idle-timeout RADIUS attribute is for it to be implemented in seconds as opposed to the minutes that the PowerConnect router uses. If this attribute is used for setting idle time instead of this configuration, the value from the idle-timeout RADIUS attribute will be converted from seconds to minutes and truncated to the nearest minute.

Filtering SSH access using ACLs

You can permit or deny SSH access to the PowerConnect using ACLs. To configure an ACL that restricts SSH access to the device, enter commands such as the following.

```
NetIron(config)# access-list 12 deny host 209.157.22.98
NetIron(config)# access-list 12 deny 209.157.23.0 0.0.0.255
NetIron(config)# access-list 12 deny 209.157.24.0/24
NetIron(config)# access-list 12 permit any
NetIron(config)# ssh access-group 12
NetIron(config)# write memory
```

Syntax: ssh access-group {<*num*> | <*name*> | **ipv6** <*ipv6-acl-name*>}

Use the **ipv6** keyword if you are applying an IPv6 access list.

The <*num*> parameter specifies the number of a standard IPv4 ACL, 1 – 99.

The <*name*> parameter specifies a standard IPv4 access list name.

The <*ipv6-acl-name*> parameter specifies an IPv6 access list name.

These commands configure ACL 12, then apply the ACL as the access list for SSH access. The device denies SSH access from the IPv4 addresses listed in ACL 12 and permits SSH access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny SSH access from all IP addresses.

NOTE
Access control lists are IP version specific. When both IPv4 and IPv6 ACLs are configured, the IPv4 ACL will be applied to sessions from IPv4 clients and the IPv6 ACL will be applied to sessions from IPv6 clients.

Refer to 21, "Access Control List" and Chapter 40, "Configuring an IPv6 Access Control List" for details on how to configure ACLs.

Disabling 3-DES

By default, both 3-DES and AES encryption algorithms are enabled on the NetIron device. You can disable 3-DES by entering the following command.

```
NetIron(config)# ip ssh encryption aes-only
```

Syntax: [no] ip ssh encryption aes-only

Displaying SSH connection information

Up to five SSH connections can be active on the PowerConnect. To display information about SSH connections, enter the following command.

```
NetIron# show ip ssh
Connection Version  Encryption    Username    IP Address
1           SSH-2   3des-cbc      Hanuma      10.43.2.4
2           SSH-2   aes128-cbc    Mikaila     10.50.3.7
3           SSH-2   aes192-cbc    Jenny       10.47.8.20
4           SSH-2   aes256-cbc    Mariah      10.55.3.9
5           SSH-2   3des-cbc      Logan       10.9.4.11
```

Syntax: show ip ssh [**| begin** <*expression*> **| exclude** <*expression*> **| include** <*expression*>]

This display shows the following information about the active SSH connections.

TABLE 388 SSH connection information

This field...	Displays...
Connection	The SSH connection ID. This can be from 1 – 16.
Version	The SSH version number. This should always be SSH-2.
Encryption	The encryption method used for the connection.
Username	The user name for the connection if password authentication is used. If public key authentication is used, username shows <*none*>.

The **show who** command also displays information about SSH connections.

Example

```
NetIron#show who
Console connections:
established, monitor enabled, in config mode
2 minutes 17 seconds in idle
Telnet connections (inbound):
1 closed
2 closed
3 closed
4 closed
5 closed
Telnet connection (outbound):
6 closed
SSH connections:
1 established, client ip address 10.43.2.4, user is hanuma
1 minutes 16 seconds in idle
2 established, client ip address 10.50.3.7, user is Mikaila
you are connecting to this session
18 seconds in idle
3 established, client ip address 10.47.8.20, user is Jenny
1 minutes 39 seconds in idle
4 established, client ip address 10.55.3.9, user is Mariah
41 seconds in idle
5 established, client ip address 10.9.4.11, user is Logan
23 seconds in idle
```

Syntax: show who [| **begin**<*expression*> | **exclude**<*expression*> | **include**<*expression*>]

Ending an SSH connection

To terminate one of the active SSH connections, enter the following command.

```
NetIron# kill ssh 1
```

Syntax: kill ssh <*connection-id*>

Using Secure Copy

Secure Copy (SCP) uses security built into SSH to transfer files between hosts on a network, providing a more secure file transfer method than Remote Copy (RCP), FTP, or TFTP. SCP automatically uses the authentication methods, encryption algorithm, and data compression level configured for SSH. For example, if password authentication is enabled for SSH, the user is prompted for a user name and password before SCP allows a file to be transferred. No additional configuration is required for SCP on top of SSH.

You can use SCP to copy files on the PowerConnect, including the startup configuration and running configuration files, to or from any other device running an SCP program (referred to here as an "SCP-enabled remote host").

SCP is enabled on the PowerConnect by default and can be disabled. To disable SCP, enter the following command.

```
NetIron(config)# ip ssh scp disable
```

Syntax: ip ssh scp disable | enable

NOTE
If you disable SSH, SCP is also disabled.

NOTE
When using SCP to transfer files, you enter the **scp** commands on the SCP-enabled remote host, rather than the console on the PowerConnect.

NOTE
Certain SCP client options, including -p and -r, are ignored by the SCP server on the PowerConnect. If an option is ignored, the client is notified.

NOTE
If password authentication is enabled for SSH, the user is prompted for user password before the file transfer takes place.

NOTE
All SCP features supported on the PowerConnect B-MLXe Series.

Outbound commands:

The following is the list of outbound SCP command options supported (Upload from device to host).

The general syntax of the outbound SCP commands is as follows.

Syntax: scp *<user>*@*<IP Address>*:*<Source>*:*<src-name> <dst-file>*

<src-name> can be abbreviated

To copy the running configuration file on a PowerConnect to a file on the SCP-enabled host.

```
C:\> scp <user>@<PowerConnect-IpAddress>:runConfig <dst-file>
```

NOTE
If you are copying the running configuration file from the PowerConnect to a PC or another machine (outbound), the command saves the running configuration file to the PC. If you are copying a configuration file from a PC to the PowerConnect, (inbound) the command appends the source file to the running configuration file on the PowerConnect.

To copy the startup configuration file on the PowerConnect to a file on the SCP-enabled client.

```
C:> scp <user>@<PowerConnect-IpAddress>:startConfig <dst-file>
```

To copy the MP primary image file from the PowerConnect to a file on the SCP-enabled client.

```
C:> scp <user>@<PowerConnect-IpAddress>:flash:primary <primary-image-file>
```

To copy the MP secondary image file from the PowerConnect to a file on the SCP-enabled client.

```
C:> scp <user>@<PowerConnect-IpAddress>:flash:secondary <secondary-image-file>
```

To copy a flash file from the PowerConnect to a file on the SCP-enabled client.

```
C:> scp <user>@<PowerConnect-IpAddress>:flash:<src-file> <dst-file>
```

To copy a file on PCMCIA slot from the PowerConnect to a file on the SCP-enabled client

```
C:> scp <user>@<PowerConnect-IpAddress>:slot1:/<src-file> <dst-file>
C:> scp <user>@<PowerConnect-IpAddress>:slot2:/<src-file> <dst-file>
```

Inbound commands:

The following is the list of inbound SCP command options supported (for downloading from an SCP-enabled client to the PowerConnect). The commands copy the files from the SCP-enabled client to the specified location on the PowerConnect.

The general syntax of the Inbound SCP commands is as follows.

Syntax: scp *<file-name>* *<user>***@***<IP Address>***:***<Destination>***:***<file-name>***[:***<additional-options>**]**

The last two tokens *<file-name>* and *<additional-options>* can be abbreviated. The others cannot be abbreviated.

PCMCIA command option

To download a file and store it in a PCMCIA (Slot 1 or Slot 2), enter the following command

```
C:> scp <src-file> <user>@<PowerConnect-IpAddress>:slot1:/<dst-file>
```

This commands transfers *<src-file>* to the PowerConnect and saves it as */slot1/<dst-file>*.

Flash MP command options

To download a file and store in MP Flash , enter the following command.

```
C:> scp <src-file> <user>@<PowerConnect-IpAddress>:flash:<dst-file>
```

This command transfers *<src-file>* to the PowerConnect and saves it as *<dst-file>* in flash

To download and replace MP Monitor image in Flash , enter the following command.

```
C:> scp <monitor-image-file> <user>@<PowerConnect-IpAddress>:flash:monitor
```

This command transfers <monitor-image-file> to the PowerConnect and replaces MP monitor image in flash.

To download and replace MP Primary image in Flash , enter the following command.

```
C:> scp <primary-image-file> <user>@<PowerConnect-IpAddress>:flash:primary
```

This command transfers *<primary-image-file>* to the PowerConnect and replaces MP Primary image in flash.

To download and replace MP secondary image in Flash , enter the following command.

```
C:> scp <secondary-image-file> <user>@<PowerConnect-IpAddress>:flash:secondary
```

This command transfers <*secondary-image-file*> to the PowerConnect and replaces MP secondary image in flash.

To download and replace MP boot image in Flash, enter the following command.

```
C:> scp <boot-image-file> <user>@<PowerConnect-IpAddress>:flash:boot
```

This command transfers <*boot-image-file*> to the PowerConnect and replaces MP boot image in flash.

Running configuration command options

To download a configuration file and append to running configuration , enter the following command.

```
C:> scp <config-file> <user>@<PowerConnect-IpAddress>:config:run
```

This command transfers <*config-file*> to the PowerConnect and appends to the running configuration.

For backward compatibility, the following syntax is also supported for this command.

```
C:> scp <config-file> <user>@<PowerConnect-IpAddress>:runConfig
```

Startup configuration command options

To download a configuration file and replace startup configuration , enter the following command.

```
C:> scp <config-file> <user>@<PowerConnect-IpAddress>:config:start
```

This command transfers *<config-file>* to the PowerConnect and replaces the startup configuration in flash.

For backward compatibility, the following syntax is also supported for this command.

```
C:> scp <config-file> <user>@<PowerConnect-IpAddress>:startConfig
```

Combined image command options

To download a combined image file and replace LP and MP primary.

```
C:> scp <combined-image-file> <user>@<PowerConnect-IpAddress>:image:primary
```

This command transfers <*combined-image-file*> to the PowerConnect and replaces MP and LP Primary image in flash.

To download a combined image file and replace LP primary and MP secondary.

```
C:> scp <combined-image-file>   <user>@<PowerConnect-IpAddress>:image:mp-sec
```

This command transfers <*combined-image-file*> to the PowerConnect and replaces MP Secondary and LP Primary image in flash.

To download a combined image file and replace LP secondary and MP primary, enter the following command.

```
C:> scp <combined-image-file> <user>@<PowerConnect-IpAddress>:image:lp-sec
```

This command transfers <*combined-image-file*> to the PowerConnect and replaces MP Primary and LP Secondary image in flash.

To download a combined image file and replace LP and MP secondary, enter the following command.

```
C:> scp <combined-image-file> <user>@<PowerConnect-IpAddress>:image:secondary
```

This command transfers <*combined-image-file*> to the PowerConnect and replaces MP and LP Secondary image in flash.

MBRIDGE command options

To download and replace FPGA (mbridge) file in MP, enter the following command.

```
C:> scp <image-file> <user>@<PowerConnect-IpAddress>:mbridge
```

This command downloads <*image-file*> and replaces the mbridge image on the flash.

Switch fabric options

To download and replace switch fabric file to a single SNM or all in MP, enter the following command.

```
C:> scp <image-file> <user>@<PowerConnect-IpAddress>:snm:sbridge:
<snm-index>
```

This command downloads <*image-file*> and replaces sbridge image on the specified SNM.

To download and replace sbridge image on all SNMs, enter the following command.

```
C:> scp <image-file> <user>@<PowerConnect-IpAddress>:snm:sbridge:all
```

This command downloads <*image-file*> and replaces the sbridge image on all the SNMs.

LP command option

To download and over-write the LP boot image on one LP or all LPs, enter the following command.

```
C:> scp <lp-boot-image-file>
<user>@<PowerConnect-IpAddress>:lp:boot:<lp-slot-num>
```

This command transfers <*lp-boot-image-file*> to the PowerConnect and replaces the LP boot image in the specified LP slot.

```
C:> scp <lp-boot-image-file> <user>@<PowerConnect-IpAddress>:lp:boot:all
```

This command transfers <*lp-boot-image-file*> to the PowerConnect and replaces LP boot image in all the LP slots

To download and over-write the LP monitor image on one LP or all LPs, enter the following command.

```
C:> scp <lp-monitor-image-file>
<user>@<PowerConnect-IpAddress>:lp:monitor:<lp-slot-num>
```

This command transfers <*lp-monitor-image-file*> to the PowerConnect and replaces the LP monitor image in the specified LP slot.

```
C:> scp <lp-monitor-image-file> <user>@<PowerConnect-IpAddress>:lp:monitor:all
```

This command transfers <*lp-monitor-image-file*> to the PowerConnect and replaces LP monitor image in all LP slots.

To download and over-write LP primary image on one LP or all LPs, enter the following commands.

```
C:> scp <lp-primary-file>
<user>@<PowerConnect-IpAddress>:lp:primary:<lp-slot-num>
```

This command transfers <lp-primary-file> to the PowerConnect and replaces the LP Primary image in the specified LP slot.

```
C:> scp <lp-primary-file> <user>@<PowerConnect-IpAddress>:lp:primary:all
```

This command transfers <*lp-primary-file*> to the PowerConnect and replaces the LP Primary image in all the LP slots.

To download and over-write the LP secondary image on one LP or all LPs, enter the following commands.

```
C:> scp <lp-secondary-file>
<user>@<PowerConnect-IpAddress>:lp:secondary:<lp-slot-num>
```

This command transfers <*lp-secondary-file*> to the PowerConnect and replaces LP Secondary image in the specified LP slot.

```
C:> scp <lp-secondary-file> <user>@<PowerConnect-IpAddress>:lp:secondary:all
```

This command transfers <*lp-secondary-file*> to the PowerConnect and replaces the LP Secondary image in all the LP slots.

Bundled FPGA command options

NOTE
If force-overwrite is present in the command, the command skips compatibility checks and forcibly replaces the FPGA image, otherwise the command checks for compatibility of the FPGA image and if the check fails the FPGA image is not not replaced and error message is returned to the SCP client.

To download and over-write Bundled FPGA image, enter the following commands.

```
C:> scp <fpga-bundle-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-all:<lp-slot-num>
```

This command downloads <*fpga-bundle-file*> and replaces all the FPGA images (PBIF, STATS, XGMAC and XPP) on the specified LP.

```
C:> scp <fpga-bundle-file> <user>@<PowerConnect-IpAddress>:lp:fpga-all:all
```

This command downloads <*fpga-bundle-file*> and replaces all the FPGA images (PBIF, STATS, XGMAC and XPP) on all the LPs.

To download and force overwrite Bundled FPGA image, enter the following.

```
C:> scp <fpga-bundle-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-all:<lp-slot-num>:force-overwrite
```

This command downloads <*fpga-bundle-file*> and replaces all the FPGA images (PBIF, STATS, XGMAC and XPP) on the specified LP.

```
C:> scp <fpga-bundle-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-all:all:force-overwrite
```

This command downloads <*fpga-bundle-file*> and replaces all the FPGA images (PBIF, STATS, XGMAC and XPP) on all the LPs.

PBIF FPGA command options

NOTE
If force-overwrite is present in the command, the command skips compatibility checks and forcibly replaces the FPGA image, otherwise the command checks for compatibility of the FPGA image and if the check fails, the FPGA image is not replaced and error message is returned to the SCP client.

To download and over-write PBIF FPGA image, enter the following command.

```
C:> scp <fpga-pbif-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-pbif:<lp-slot-num>
```

This command downloads <*fpga-pbif-file*> and replaces the FPGA PBIF image on the specified LP.

```
C:> scp <fpga-pbif-file> <user>@<PowerConnect-IpAddress>:lp:fpga-pbif:all
```

This command downloads <*fpga-pbif-file*> and replaces FPGA PBIF image on all the LPs.

To download and force over-write PBIF FPGA image, enter the following command.

```
C:> scp <fpga-pbif-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-pbif:<lp-slot-num>:force-overwrite
```

This command downloads <*fpga-pbif-file*> and replaces FPGA PBIF image on the specified LP.

```
C:> scp <fpga-pbif-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-pbif:all:force-overwrite
```

This command downloads <*fpga-pbif-file*> and replaces the FPGA PBIF image on all the LPs.

STATS FPGA command options

NOTE
If force-overwrite is present in the command, the command skips compatibility checks and forcibly replaces the FPGA image, otherwise the command checks for compatibility of the FPGA image and if the check fails, the FPGA image is not replaced and error message is returned to the SCP client.

To download and over-write STATS FPGA image, enter the following.

```
C:> scp <fpga-stats-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-stats:<lp-slot-num>
```

This command downloads <*fpga-stats-file*> and replaces FPGA STATS image on the specified LP.

```
C:> scp <fpga-stats-file> <user>@<PowerConnect-IpAddress>:lp:fpga-stats:all
```

This command downloads <*fpga-stats-file*> and replaces FPGA STATS image on all the LPs.

To download and force over-write STATS FPGA image, enter the following command.

```
C:> scp <fpga-stats-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-stats:<lp-slot-num>:force-overwrite
```

This command downloads <*fpga-stats-file*> and replaces FPGA STATS image on the specified LP.

```
C:> scp <fpga-stats-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-stats:all:force-overwrite
```

This command downloads <*fpga-stats-file*> and replaces FPGA STATS image on all the LPs.

XGMAC FPGA command options

NOTE
If force-overwrite is present in the command, the command skips compatibility checks and forcibly replaces the FPGA image, otherwise the command checks for compatibility of the FPGA image and if the check fails, the FPGA image is not replaced and error message is returned to the SCP client.

To download and over-write XGMAC FPGA image, enter the following commands.

```
C:> scp <fpga-xgmac-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-xgmac:<lp-slot-num>
```

This command downloads <*fpga-xgmac-file*> and replaces FPGA XGMAC image on the specified LP.

```
C:> scp <fpga-xgmac-file> <user>@<PowerConnect-IpAddress>:lp:fpga-xgmac:all
```

This command downloads <*fpga-xgmac-file*> and replaces FPGA XGMAC image on all the LPs.

To download and force over-write XGMAC FPGA image, enter the following commands.

```
C:> scp <fpga-xgmac-file> <user>@<PowerConnect-IpAddress>:lp:
fpga-xgmac:<lp-slot-num>:force-overwrite
```

This command downloads <*fpga-xgmac-file*> and replaces FPGA XGMAC image on the specified LP.

```
C:> scp <fpga-xgmac-file>
<user>@<PowerConnect-IpAddress>:lp:fpga-xgmac:all:force-overwrite
```

This command downloads <*fpga-xgmac-file*> and replaces FPGA XGMAC image on all the LPs.

XPP FPGA command options

NOTE
If force-overwrite is present in the command, the command skips compatibility checks and forcibly replaces the FPGA image, otherwise the command checks for compatibility of the FPGA image and if the check fails, the FPGA image is not replaced and error message is returned to the SCP client.

To download and over-write an XPP FPGA image, enter the following commands.

```
C:> scp <fpga-xpp-file> <user>@<PowerConnect-IpAddress>:lp:fpga-xpp:<lp-slot-num>
```

This command downloads <*fpga-xpp-file*> and replaces FPGA XPP image on the specified LP.

```
C:> scp <fpga-xpp-file> <user>@<PowerConnect-IpAddress>:lp:fpga-xpp:all
```

This command downloads <*fpga-xpp-file*> and replaces FPGA XPP image on all the LPs.

To download and force over-write XPP FPGA image, enter the following commands.

```
C:> scp <fpga-xpp-file> <user>@<PowerConnect-IpAddress>:lp:fpga-xpp:<
lp-slot-num>:force-overwrite
```

This command downloads <*fpga-xpp-file*> and replaces FPGA XPP image on the specified LP.

```
C:> scp <fpga-xpp-file> <user>@<MLXe-IpAddress>:lp:fpga-xpp:all:force-overwrite
```

This command downloads <*fpga-xpp-file*> and replaces FPGA XPP image on all the LPs.

Delete old file first option

NOTE
The delete file first option only applies to inbound SCP commands; its purpose is make room in the MP flash by deleting old image files prior to an image download.

An option "delete-first" is provided in the third or fourth token position in the following commands."

```
C:> scp <image-file>
<user>@<PowerConnect-IpAddress>:image:<primary|secondary|mp-sec|lp-sec>:delete-first
C:> scp <image-file>
<user>@<PowerConnect-IpAddress>:flash:<primary|secondary|monitor>:delete-first
C:> scp <image-file>
<user>@<PowerConnect-IpAddress>:lp:<primary|secondary|monitor>:all:delete-first
```

Without **delete-first** option, if the flash is full these commands should display the following message.

"There is not enough space on MP flash. Please clean up MP flash and retry, or use \"delete-first\" option."

When the **delete-first** option is specified, these commands clear space in the MP flash by removing the following files first.

```
image:primary, "primary", "lp-primary-0"
image:secondary, "secondary", "lp-secondary-0"
image:lp-sec, "primary", "lp-secondary-0"
image:mp-sec, "secondary", "lp-primary-0"
flash:primary, "primary"
flash: secondary, "secondary"
flash: monitor, "monitor"
lp:primary:all, "lp-primary-0"
lp:secondary:all, "lp-secondary-0"
lp:monitor:all, "lp-monitor-0"
```

Before deleting the file the system will check to see if deleting the file or files will create enough space in the flash. If it can create enough space to accommodate the download, the files will be deleted. Otherwise, the command will fail with the following error message.

```
"There will not be enough space on MP flash even after deleting the target files.
Please clean up MP flash and retry."
```

NOTE
Other commands will not check for available space in the flash or delete the file before downloading. In other words, the delete-first option is not supported for commands not mentioned above.

Chapter

Configuring Multi-Device Port Authentication 49

The following Multi-Device Port Authenticationt features supported by PowerConnect B-MLXe Series.

- Multi-Device Port Authentication
- Authentication Method List for 802.1x
- RADIUS Parameters
- Authentication-Failure Action
- Authenticated MAC Addresses
- MAC Address Filters
- Authenticating Multiple MAC Addresses on an Interface
- Multi-Device Port Authentication and 802.1x on the Same Interface
- Aging Time for Blocked MAC Addresses

Multi-device port authentication is a way to configure a PowerConnect router to forward or block traffic from a MAC address based on information received from a RADIUS server.

How multi-device port authentication works

The multi-device port authentication feature is a mechanism by which incoming traffic originating from a specific MAC address is switched or forwarded by the device only if the source MAC address is successfully authenticated by a RADIUS server. The MAC address itself is used as the username and password for RADIUS authentication; the user does not need to provide a specific username and password to gain access to the network. If RADIUS authentication for the MAC address is successful, traffic from the MAC address is forwarded in hardware.

If the RADIUS server cannot validate the user's MAC address, then it is considered an authentication failure, and a specified authentication-failure action can be taken. The default authentication-failure action is to drop traffic from the non-authenticated MAC address in hardware. You can also configure the device to move the port on which the non-authenticated MAC address was learned into a restricted or “guest” VLAN, which may have limited access to the network.

RADIUS authentication

The multi-device port authentication feature communicates with the RADIUS server to authenticate a newly found MAC address. The PowerConnect devices support multiple RADIUS servers; if communication with one of the RADIUS servers times out, the others are tried in sequential order. If a response from a RADIUS server is not received within a specified time (by default, 3 seconds) the RADIUS session times out, and the device retries the request up to three times. If no response is received, the next RADIUS server is chosen, and the request is sent for authentication.

The RADIUS server is configured with the usernames and passwords of authenticated users. For multi-device port authentication, the username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server. For example, given a MAC address of 0007e90feaa1, the users file on the RADIUS server would be configured with a username and password both set to 0007e90feaa1. When traffic from this MAC address is encountered on a MAC-authentication-enabled interface, the device sends the RADIUS server an Access-Request message with 0007e90feaa1 as both the username and password. The format of the MAC address sent to the RADIUS server is configurable through the CLI.

The request for authentication from the RADIUS server is successful only if the username and password provided in the request matches an entry in the users database on the RADIUS server. When this happens, the RADIUS server returns an Access-Accept message back to the PowerConnect router. When the RADIUS server returns an Access-Accept message for a MAC address, that MAC address is considered authenticated, and traffic from the MAC address is forwarded normally by the PowerConnect router.

Authentication-failure actions

If the MAC address does not match the username and password of an entry in the users database on the RADIUS server, then the RADIUS server returns an Access-Reject message. When this happens, it is considered an authentication failure for the MAC address. When an authentication failure occurs, the PowerConnect devices can either drop traffic from the MAC address in hardware (the default), or move the port on which the traffic was received to a restricted VLAN.

PowerConnect devices support multi-device port authentication on untagged ports only.

Supported RADIUS attributes

The PowerConnect devices support the following RADIUS attributes for multi-device port authentication:

- Username (1) – RFC 2865
- FilterId (11) – RFC 2865
- Vendor-Specific Attributes (26) – RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 3579
- Tunnel-Private-Group-Id (81) – RFC 2868

Dynamic VLAN and ACL assignments

The multi-device port authentication feature supports ***dynamic VLAN assignment***, where a port can be placed in a VLAN based on the MAC address learned on that interface. When a MAC address is successfully authenticated, the RADIUS server sends the PowerConnect router a RADIUS Access-Accept message that allows the PowerConnect router to forward traffic from that MAC address. The RADIUS Access-Accept message can also contain attributes set for the MAC address in its access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies a VLAN identifier, and this VLAN is available on the PowerConnect router, the port is moved from its default VLAN to the specified VLAN.

To enable dynamic VLAN assignment for authenticated MAC addresses, you must add the following attributes to the profile for the MAC address on the RADIUS server. Dynamic VLAN assignment on multi-device port authentication-enabled interfaces is enabled by default.

Attribute name	Type	Value
Tunnel-Type	064	13 (decimal) – VLAN
Tunnel-Medium-Type	065	6 (decimal) – 802
Tunnel-Private-Group-ID	081	<*vlan-name*> (string) – either the name or the number of a VLAN configured on the PowerConnect device.

In addition to dynamic VLAN assignment, PowerConnect devices also support dynamic ACL assignment as is the case with 802.1x port security.

Support for authenticating multiple MAC addresses on an interface

The multi-device port authentication feature allows multiple MAC addresses to be authenticated or denied authentication on each interface. The maximum number of MAC addresses that can be authenticated on each interface is 256. The default is 32.

Support for multi-device port authentication and 802.1x on the same interface

On the PowerConnect devices, multi-device port authentication and 802.1x security can be enabled on the same port. However, only one of them can authenticate a MAC address or 802.1x client. If an 802.1x client responds, the software assumes that the MAC should be authenticated using 802.1x protocol mechanisms and multi-device port authentication for that MAC is aborted. Also, at any given time, a port can have either 802.1x clients or multi-device port authentication clients but not both.

Configuring multi-device port authentication

Configuring multi-device port authentication on the PowerConnect devices consists of the following tasks:

- Enabling multi-device port authentication globally and on individual interfaces
- Configuring an Authentication Method List for 802.1x
- Setting RADIUS Parameters
- Specifying the format of the MAC addresses sent to the RADIUS server (optional)
- Specifying the authentication-failure action (optional)
- Defining MAC address filters (optional)
- Configuring dynamic VLAN assignment (optional)

- Specifying to which VLAN a port is moved after its RADIUS-specified VLAN assignment expires (optional)
- Saving dynamic VLAN assignments to the running configuration file (optional)
- Clearing authenticated MAC addresses (optional)
- Disabling aging for authenticated MAC addresses (optional)
- Specifying the aging time for blocked MAC addresses (optional)

Enabling multi-device port authentication

You globally enable multi-device port authentication on the PowerConnect router.

To globally enable multi-device port authentication on the device, enter the following command.

```
NetIron(config)# mac-authentication enable
```

Syntax: [no] mac-authentication enable

Syntax: [no] mac-authentication enable <*slot*>/<*portnum*> **| all**

The **all** option enables the feature on all interfaces at once.

You can enable the feature on an interface at the interface CONFIG level.

Configuring an authentication method list for 802.1x

To use 802.1x port security, you must specify an authentication method to be used to authenticate Clients. The PowerConnect device supports RADIUS authentication with 802.1x port security. To use RADIUS authentication with 802.1x port security, you create an authentication method list for 802.1x and specify RADIUS as an authentication method, then configure communication between the PowerConnect router and the RADIUS server.

Example

```
NetIron(config)# aaa authentication dot1x default radius
```

Syntax: [no] aaa authentication dot1x default <*method-list*>

For the <*method-list*>, enter at least one of the following authentication methods:

radius – Use the list of all RADIUS servers that support 802.1x for authentication.

none – Use no authentication. The Client is automatically authenticated without the device using information supplied by the Client.

NOTE
If you specify both **radius** and **none**, make sure **radius** comes before **none** in the method list.

Setting RADIUS parameters

To use a RADIUS server to authenticate access to a PowerConnect router, you must identify the server to the PowerConnect router.

Example

```
NetIron(config)# radius-server host 209.157.22.99 auth-port 1812 acct-port 1813
default key mirabeau dot1x
```

Syntax: **radius-server host** <*ip-addr*> **|** <*server-name*> **[auth-port** <*number*> **acct-port** <*number*> **[authentication-only | accounting-only | default [key 0 | 1** <*string*> **[dot1x]]]]**

The **host** <*ip-addr*> | <*server-name*> parameter is either an IP address or an ASCII text string.

The **auth-port** <*number*> parameter specifies what port to use for RADIUS authentication.

The **acct-port** <*number*> parameter specifies what port to use for RADIUS accounting.

The **dot1x** parameter indicates that this RADIUS server supports the 802.1x standard. A RADIUS server that supports the 802.1x standard can also be used to authenticate non-802.1x authentication requests.

NOTE
To implement 802.1x port security, at least one of the RADIUS servers identified to the router must support the 802.1x standard.

Supported RADIUS attributes

Many IEEE 802.1x Authenticators will function as RADIUS clients. Some of the RADIUS attributes may be received as part of IEEE 802.1x authentication. The PowerConnect devices support the following RADIUS attributes for IEEE 802.1x authentication:

- Username (1) – RFC 2865
- FilterId (11) – RFC 2865
- Vendor-Specific Attributes (26) – RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 2579
- Tunnel-Private-Group-Id (81) – RFC 2868

Specifying the format of the MAC addresses sent to the RADIUS server

When multi-device port authentication is configured, the PowerConnect router authenticates MAC addresses by sending username and password information to a RADIUS server. The username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server.

By default, the MAC address is sent to the RADIUS server in the format *xxxxxxxxxxxx*. You can optionally configure the device to send the MAC address to the RADIUS server in the format *xx-xx-xx-xx-xx-xx*, or the format *xxxx.xxxx.xxxx*. To do this, enter a command such as the following.

```
NetIron(config)# mac-authentication auth-passwd-format xxxx.xxxx.xxxx
```

Syntax: **[no] mac-authentication auth-passwd-format xxxx.xxxx.xxxx | xx-xx-xx-xx-xx-xx | xxxxxxxxxxxx**

Specifying the authentication-failure action

When RADIUS authentication for a MAC address fails, you can configure the device to perform one of two actions:

- Drop traffic from the MAC address in hardware (the default)
- Move the port on which the traffic was received to a restricted VLAN

To configure the device to move the port to a restricted VLAN when multi-device port authentication fails, enter commands such as the following.

```
NetIron(config)# interface e 3/1
NetIron(config-if-e100-3/1)# mac-authentication auth-fail-action restrict-vlan
100
```

Syntax: [no] mac-authentication auth-fail-action restrict-vlan [<*vlan-id*>**]**

If the ID for the restricted VLAN is not specified at the interface level, the global restricted VLAN ID applies for the interface.

To specify the VLAN ID of the restricted VLAN globally, enter the following command.

```
NetIron(config)# mac-authentication auth-fail-vlan-id 200
```

Syntax: [no] mac-authentication auth-fail-vlan-id <*vlan-id*>

The command above applies globally to all MAC-authentication-enabled interfaces.

Note that the restricted VLAN must already exist on the device. You cannot configure the restricted VLAN to be a non-existent VLAN.

To configure the device to drop traffic from non-authenticated MAC addresses in hardware, enter commands such as the following.

```
NetIron(config)# interface e 3/1
NetIron(config-if-e100-3/1)# mac-authentication auth-fail-action block-traffic
```

Syntax: [no] mac-authentication auth-fail-action block-traffic

Dropping traffic from non-authenticated MAC addresses is the default behavior when multi-device port authentication is enabled.

Defining MAC address filters

You can specify MAC addresses that do not have to go through multi-device port authentication. These MAC addresses are considered pre-authenticated, and are not subject to RADIUS authentication. To do this, you can define MAC address filters that specify the MAC addresses to exclude from multi-device port authentication.

You should use a MAC address filter when the RADIUS server itself is connected to an interface where multi-device port authentication is enabled. If a MAC address filter is not defined for the MAC address of the RADIUS server and applied on the interface, the RADIUS authentication process would fail since the device would drop all packets from the RADIUS server itself.

For example, the following command defines a MAC address filter for address 0010.dc58.aca4.

```
NetIron(config)# mac-authentication mac-filter 1 permit 0010.dc58.aca4
```

Syntax: [no] mac-authentication mac-filter <*filter*>

The following commands apply the MAC address filter on an interface so that address 0010.dc58.aca4 is excluded from multi-device port authentication.

```
NetIron(config)# interface e 3/1
NetIron(config-if-e100-3/1)# mac-authentication apply-mac-auth-filter 1
```

Syntax: [no] mac-authentication apply-mac-auth-filter <*filter-id*>

Configuring dynamic VLAN assignment

An interface can be dynamically assigned to a VLAN based on the MAC address learned on that interface. When a MAC address is successfully authenticated, the RADIUS server sends the PowerConnect router a RADIUS Access-Accept message that allows the PowerConnect router to forward traffic from that MAC address. The RADIUS Access-Accept message can also contain attributes set for the MAC address in its access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies a VLAN identifier, and this VLAN is available on the PowerConnect router, the port is moved from its default VLAN to the specified VLAN.

To enable dynamic VLAN assignment for authenticated MAC addresses, you must add the following attributes to the profile for the MAC address on the RADIUS server (dynamic VLAN assignment on multi-device port authentication-enabled interfaces is enabled by default and can be disabled). Refer to "Dynamic VLAN and ACL assignments" on page 2036 for a list of the attributes that must be set on the RADIUS server

Dynamic VLAN assignment on a multi-device port authentication-enabled interface is enabled by default. If it is disabled, enter commands such as the following command to enable it.

```
NetIron(config)# interface e 3/1
NetIron(config-if-e100-3/1)# mac-authentication enable-dynamic-vlan
```

Syntax: [no] mac-authentication enable-dynamic-vlan

If a previous authentication attempt for a MAC address failed, and as a result the port was placed in the restricted VLAN, but a subsequent authentication attempt was successful, the RADIUS Access-Accept message may specify a VLAN for the port. By default, the PowerConnect router moves the port out of the restricted VLAN and into the RADIUS-specified VLAN. You can optionally configure the device to ignore the RADIUS-specified VLAN in the RADIUS Access-Accept message, and leave the port in the restricted VLAN.

To do this, enter the following command.

```
NetIron(config)# mac-authentication no-override-restrict-vlan
```

Syntax: [no] mac-authentication no-override-restrict-vlan

NOTES:

- For untagged ports, if the VLAN ID provided by the RADIUS server is valid, then the port is removed from its current VLAN and moved to the RADIUS-specified VLAN as an untagged port.
- If you configure dynamic VLAN assignment on a multi-device port authentication enabled interface, and the Access-Accept message returned by the RADIUS server does not contain a Tunnel-Private-Group-ID attribute, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.

- If the <*vlan-name*> string does not match either the name or the ID of a VLAN configured on the device, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.
- If an untagged port had previously been assigned to a VLAN though dynamic VLAN assignment, and then another MAC address is authenticated on the same port, but the RADIUS Access-Accept message for the second MAC address specifies a different VLAN, then it is considered an authentication failure for the second MAC address, and the configured authentication failure action is performed. Note that this applies only if the first MAC address has not yet aged out. If the first MAC address has aged out, then dynamic VLAN assignment would work as expected for the second MAC address.

Specifying the VLAN to which a port is moved after the RADIUS-specified VLAN assignment expires

When a port is dynamically assigned to a VLAN through the authentication of a MAC address, and the MAC session for that address is deleted on the PowerConnect router, then by default the port is removed from its RADIUS-assigned VLAN and placed back in the VLAN where it was originally assigned.

A port can be removed from its RADIUS-assigned VLAN when any of the following occur:

- The link goes down for the port
- The MAC session is manually deleted with the **mac-authentication clear-mac-session** command
- The MAC address that caused the port to be dynamically assigned to a VLAN ages out

For example, say port 1/1 is currently in VLAN 100, to which it was assigned when MAC address 0007.eaa1.e90f was authenticated by a RADIUS server. The port was originally configured to be in VLAN 111. If the MAC session for address 0007.eaa1.e90f is deleted, then port 1/1 is moved from VLAN 100 back into VLAN 111.

You can optionally specify an alternate VLAN to which to move the port when the MAC session for the address is deleted. For example, to place the port in the restricted VLAN, enter commands such as the following.

```
NetIron(config)# interface e 3/1
NetIron(config-if-e100-3/1)# mac-auth move-back-to-old-vlan port-restrict-vlan
```

Syntax: **[no] mac-authentication move-back-to-old-vlan disable | port-configured-vlan | port-restrict-vlan | system-default-vlan**

The **disable** keyword disables moving the port back to its original VLAN. The port would stay in its RADIUS-assigned VLAN.

The **port-configured-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it back in the VLAN where it was originally assigned. This is the default.

The **port-restrict-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it in the restricted VLAN.

The **system-default-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it in the DEFAULT-VLAN.

Saving dynamic VLAN assignments to the running configuration file

You can configure the PowerConnect router to save the RADIUS-specified VLAN assignments to the device's running configuration file. To do this, enter the following command.

```
NetIron(config)# mac-authentication save-dynamicvlan-to-config
```

Syntax: [no] mac-authentication save-dynamicvlan-to-config

By default, the dynamic VLAN assignments are not saved to the running configuration file. Entering the **show running-config** command does not display dynamic VLAN assignments, although they can be displayed with the **show vlan** and **show auth-mac-address detail** commands.

Clearing authenticated MAC addresses

The PowerConnect router maintains an internal table of the authenticated MAC addresses (viewable with the **show authenticated-mac-address** command). You can clear the contents of the authenticated MAC address table either entirely, or just for the entries learned on a specified interface. In addition, you can clear the MAC session for an address learned on a specific interface.

To clear the entire contents of the authenticated MAC address table, enter the following command.

```
NetIron(config)# clear auth-mac-table
```

Syntax: clear auth-mac-table

To clear the authenticated MAC address table of entries learned on a specified interface, enter a command such as the following.

```
NetIron(config)# clear auth-mac-table e 3/1
```

Syntax: clear auth-mac-table *<slot>/<portnum>*

To clear the MAC session for an address learned on a specific interface, enter commands such as the following.

```
NetIron(config)# interface e 3/1
NetIron(config-if-e100-3/1)# mac-authentication clear-mac-session 00e0.1234.abd4
```

Syntax: mac-authentication clear-mac-session *<mac-address>*

This command removes the Layer 2 CAM entry created for the specified MAC address. If the PowerConnect router receives traffic from the MAC address again, the MAC address is authenticated again.

Disabling aging for authenticated MAC addresses

MAC addresses that have been authenticated or denied by a RADIUS server are aged out if no traffic is received from the MAC address for a certain period of time:

- Authenticated MAC addresses or non-authenticated MAC addresses that have been placed in the restricted VLAN are aged out if no traffic is received from the MAC address over the device's normal MAC aging interval.
- Non-authenticated MAC addresses that are blocked by the device are aged out if no traffic is received from the address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. (Refer to the next section for more information on configuring the software aging period).

You can optionally disable aging for MAC addresses subject to authentication, either for all MAC addresses or for those learned on a specified interface.

To disable aging for all MAC addresses subject to authentication on all interfaces where multi-device port authentication has been enabled, enter the following command.

```
NetIron(config)# mac-authentication disable-aging
```

To disable aging for all MAC addresses subject to authentication on a specific interface where multi-device port authentication has been enabled, enter commands such as the following.

```
NetIron(config)# interface e 3/1
NetIron(config-if-e100-3/1)# mac-authentication disable-aging
```

Syntax: [no] mac-authentication disable-aging [denied-mac-only | permitted-mac-only]

denied-mac-only disables aging of denied sessions and enables aging of permitted sessions.

permitted-mac-only disables aging of permitted (authenticated and restricted) sessions and enables aging of denied sessions.

Specifying the aging time for blocked MAC addresses

When the PowerConnect router is configured to drop traffic from non-authenticated MAC addresses, traffic from the blocked MAC addresses is dropped in hardware, without being sent to the CPU. A Layer 2 CAM entry is created that drops traffic from the blocked MAC address in hardware. If no traffic is received from the blocked MAC address for a certain amount of time, this Layer 2 CAM entry is aged out. If traffic is subsequently received from the MAC address, then an attempt can be made to authenticate the MAC address again.

Aging of the Layer 2 CAM entry for a blocked MAC address occurs in two phases, known as ***hardware aging*** and ***software aging***. The hardware aging period is fixed at 70 seconds and is non-configurable. The software aging time is configurable through the CLI.

Once the PowerConnect router stops receiving traffic from a blocked MAC address, the hardware aging begins and lasts for a fixed period of time. After the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (by default 120 seconds). After the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the PowerConnect router receives traffic from the MAC address.

To change the length of the software aging period for blocked MAC addresses, enter a command such as the following.

```
NetIron(config)# mac-authentication max-age 180
```

Syntax: [no] mac-authentication max-age <*seconds*>

You can specify from 1 – 65535 seconds. The default is 120 seconds.

Displaying multi-device port authentication information

You can display the following information about the multi-device port authentication configuration:

- Information about authenticated MAC addresses
- Information about the multi-device port authentication configuration

- Authentication Information for a specific MAC address or port
- Multi-device port authentication settings and authenticated MAC addresses for each port where the multi-device port authentication feature is enabled
- The MAC addresses that have been successfully authenticated
- The MAC addresses for which authentication was not successful

Displaying authenticated MAC address information

To display information about authenticated MAC addresses on the ports where the multi-device port authentication feature is enabled, enter the following command.

```
NetIron# show auth-mac-address
------------------------------------------------------------------------
Port           Vlan  Accepted MACs    Rejected MACs    Attempted-MACs
------------------------------------------------------------------------
1/18           100   1                  100              0
1/20           40    0                  0                0
1/22           100   0                  0                0
4/5            30    0                  0                0
```

Syntax: show auth-mac-address

The following table describes the information displayed by the **show auth-mac-address** command.

TABLE 389 Output from the **show auth-mac-address command**

This field...	Displays...
Port	The port number where the multi-device port authentication feature is enabled.
Vlan	The VLAN to which the port has been assigned.
Accepted MACs	The number of MAC addresses that have been successfully authenticated
Rejected MACs	The number of MAC addresses for which authentication has failed.
Attempted-MACs	The rate at which authentication attempts are made for MAC addresses.

Displaying multi-device port authentication configuration information

To display a summary of multi-device port authentication that have been configured on the device, enter the following command.

```
NetIron# show auth-mac configuration
Feature enabled                 : Yes
Global Fail-VLAN Id             : None
Username/Password format        : xxxx.xxxx.xxxx
Maximum Age                     : 120
Save dynamic VLAN configuration : No
Number of Ports enabled         : 25
-------------------------------------------------------------------------------
Port  Aging     Fail       Fail DynVLAN Override   Revert      MAC    DoS Protectn
                Action     VLAN Support Restricted VLAN        Filter Enable Limit
-------------------------------------------------------------------------------
1/1   All       Blocked    N/A  Yes     Yes        Configured  No     No     512
1/2   Permitted Blocked    101  No      Yes        Restricted  No     No     512
1/3   All       Blocked    N/A  Yes     Yes        Configured  No     No     512
1/4   Denied    Blocked    N/A  Yes     Yes        Configured  No     No     512
1/5   All       Blocked    N/A  Yes     Yes        Configured  No     No     512
1/6   None      Blocked    N/A  Yes     Yes        Sys.Default No     No     512
1/7   All       Blocked    N/A  Yes     Yes        Configured  No     No     512
1/8   All       Blocked    N/A  Yes     Yes        Configured  No     No     512
1/9   All       Blocked    N/A  Yes     Yes        Configured  No     No     512
1/10  All       Blocked    N/A  Yes     Yes        Configured  No     No     512
```

The following table describes the information displayed by the **show authenticated-mac-address configuration** command.

TABLE 390 Output from the **show auth-mac-addre**ss configuration c**ommand**

This field...	Displays...
Feature enabled	Whether the multi-device port authentication feature is enabled on the PowerConnect device.
Number of Ports enabled	The number of ports on which the multi-device port authentication feature is enabled.
Aging	Shows which MAC addresses are aged out. Denied – Only denied MAC addresses are aged out Permitted – Only permitted MAC addresses are aged out All – Both denied and permitted MAC addresses are aged out None – None of the MAC addresses are aged out
Port	Information for each multi-device port authentication-enabled port.
Fail-Action	What happens to traffic from a MAC address for which RADIUS authentication has failed: either block the traffic or assign the MAC address to a restricted VLAN.
Fail VLAN	The restricted VLAN to which non-authenticated MAC addresses are assigned, if the Fail-Action is to assign the MAC address to a restricted VLAN.
DynVLAN Support	Whether RADIUS dynamic VLAN assignment is enabled for the port.
Override Restricted	Whether or not a port in a restricted VLAN (due to a failed authentication) is removed from the restricted VLAN on a subsequent successful authentication on the port.

TABLE 390 Output from the **show auth-mac-address** configuration **command** (Continued)

This field...	Displays...
Revert VLAN	The VLAN that the port reverts to when the RADIUS-assigned dynamic VLAN expires.
MAC-filter	Whether a MAC filter has been applied to this port to specify pre-authenticated MAC addresses.
DOS Enable	Denial of Service status. This column will always show "No" since DOS is not supported.
Protectn Limit	This is not applicable to the PowerConnect, but the output always show "512".

Syntax: show auth-mac-address configuration

To display detailed information about the multi-device port authentication configuration and authenticated MAC addresses for a port where the feature is enabled, enter the following command.

```
NetIron# show auth-mac-address detail
Port 1/18
 Dynamic-Vlan Assignment          : Enabled
 RADIUS failure action            : Block Traffic
 Override-restrict-vlan           : Yes
 Port VLAN                        : 4090 (Configured)
 DOS attack protection            : Disabled
 Accepted Mac Addresses           : 0
 Rejected Mac Addresses           : 0
 Aging of MAC-sessions            : Enable-All
 Port move-back vlan              : Port-Configured
 MAC Filter applied               : No
                                1 : 0000.0010.2000

MAC TABLE
-------------------------------------------
MAC Address    Port    VLAN Access     Age

-------------------------------------------
00A1.0010.2000 1/18    1    Allowed    0
00A1.0010.2001 1/18    1    Blocked    120
00A1.0010.2002 1/18    1    Init       0
```

The following table describes the information displayed by the **show authenticated-mac-address** command.

TABLE 391 Output from the **show authenticated-mac-address** c**ommand**

This field...	Displays...
Port	The port to which this information applies.
Dynamic-Vlan Assignment	Whether RADIUS dynamic VLAN assignment has been enabled for the port.
RADIUS failure action	What happens to traffic from a MAC address for which RADIUS authentication has failed: either block the traffic or assign the MAC address to a restricted VLAN.
Override-restrict-vlan	Whether a port can be dynamically assigned to a VLAN specified by a RADIUS server, if the port had been previously placed in the restricted VLAN because a previous attempt at authenticating a MAC address on that port failed.

TABLE 391 Output from the **show authenticated-mac-address command** (Continued)

This field...	Displays...
Port VLAN	The VLAN to which the port is assigned, and whether the port had been dynamically assigned to the VLAN by a RADIUS server.
DOS attack protection	Whether denial of service attack protection has been enabled for multi-device port authentication, limiting the rate of authentication attempts sent to the RADIUS server.
Accepted MAC Addresses	The number of MAC addresses that have been successfully authenticated.
Rejected MAC Addresses	The number of MAC addresses for which authentication has failed.
Aging of MAC-sessions	Whether software aging of MAC addresses is enabled.
Max-Age of MAC-sessions	The configured software aging period.
Port move-back VLAN	The VLAN that the port reverts to when the RADIUS-assigned dynamic VLAN expires.
MAC Filter applied	Whether a MAC filter has been applied to this port to specify pre-authenticated MAC addresses.
MAC Table	The MAC addresses learned on the port.

Syntax: show auth-mac-address detail

Displaying multi-device port authentication information for a specific MAC address or port

To display authentication information for a specific MAC address or port, enter a command such as the following.

```
NetIron# show auth-mac-address 0007.e90f.eaa1
--------------------------------------------------------------------------------
MAC/IP Address     Port        Vlan    Access      Age
--------------------------------------------------------------------------------
00A1.0010.2000     1/18        1       Allowed     0
```

Syntax: show auth-mac-address <*mac-address*> | <*ip-address*> | <*slot*>/<*portnum*>

The <*ip-address*> parameter lists the MAC address associated with the specified IP address.

The <*slot*>/<*portnum*> parameter lists the MAC addresses on the specified port.

The following table describes the information displayed by the **show auth-mac-address** command for a specified MAC address or port.

TABLE 392 Output from the **show auth-mac-address** <*address*> **command**

This field...	Displays...
MAC or IP Address	The MAC address for which information is displayed. If the packet for which multi-device port authentication was performed also contained an IP address, then the IP address is displayed as well.
Port	The port on which the MAC address was learned.
VLAN	The VLAN to which the MAC address was assigned.

TABLE 392 Output from the **show auth-mac-address** <*address*> **command** (Continued)

This field...	Displays...
Access	Whether or not the MAC address was allowed or denied access into the network.
Age	The age of the MAC address entry in the authenticated MAC address list.

Displaying the authenticated MAC addresses

To display the MAC addresses that have been successfully authenticated, enter the following command.

```
NetIron# show auth-mac-addresses authorized-mac
MAC TABLE
-------------------------------------------
MAC Address    Port    VLAN Access     Age

-------------------------------------------
00A1.0010.2000 1/18    1    Allowed      0
00A1.0010.2001 1/18    1    Allowed    120
00A1.0010.2002 1/18    1    Allowed      0
```

Syntax: show auth-mac-addresses authorized-mac

Displaying the non-authenticated MAC addresses

To display the MAC addresses for which authentication was not successful, enter the following command.

```
NetIron# show auth-mac-addresses unauthorized-mac
MAC TABLE
-------------------------------------------
MAC Address    Port    VLAN Access     Age

-------------------------------------------
00A1.0010.2000 1/18    1    Blocked       0
00A1.0010.2001 1/18    1    Blocked     120
00A1.0010.2002 1/18    1    Blocked       0
```

Syntax: show auth-mac-addresses unauthorized-mac

Chapter

Using the MAC Port Security Feature 50

The following MAC Port Security features are supported by PowerConnect B-MLXe Series.

- MAC Port Security
- Port Security Age Timer
- Denying Specific MAC Addresses
- Port Security MAC Violation Limit
- MAC Port Security on VPLS endpoints
- MAC Port Security on Vll endpoints

Overview

MAC Port Security allows you to configure the PowerConnect router to learn a limited number of "secure" MAC addresses on an interface. The interface will forward only packets with source MAC addresses that match these secure addresses. The secure MAC addresses can be specified manually, or the PowerConnect router can learn them automatically. After the device reaches the limit for the number of secure MAC addresses it can learn on the interface, if the interface then receives a packet with a source MAC address that is different from any of the secure learned addresses, it is considered a security violation.

When a security violation occurs, a Syslog entry and an SNMP trap are generated. In addition, the device takes one of two actions: it either drops packets from the violating address (but allows packets from the secure addresses), or it disables the port for a specified amount of time. You specify which of these actions takes place.

The secure MAC addresses are not flushed when an interface is disabled and brought up again. The secure addresses can be kept secure permanently (the default), or can be configured to age out, at which time they are no longer secure. You can configure the device to automatically save the list of secure MAC addresses to the startup-config file at specified intervals, allowing addresses to be kept secure across system restarts.

The port security feature applies only to Ethernet interfaces.

Configuration Considerations

When using the MAC port security feature, the following should be considered.

- If there is no port security configuration at the interface level, global level port security configuration is inherited.
- If a port security attribute is configured at the interface level, interface level configuration for that attribute takes precedence over global level configuration for the same attribute.The rest of the port security attributes that are not configured at the interface level will be inherited from the global level configuration.

Local and global resources

The port security feature uses a concept of local and global “resources” to determine how many MAC addresses can be secured on each interface. In this context, a “resource” is the ability to store one secure MAC address entry. Each interface is allocated 64 local resources. When the port security feature is enabled, the interface can store up to 64 secure MAC addresses using local resources.

Besides the maximum of 64 local resources available to an interface, there are 4096 global resources available. When an interface has secured enough MAC addresses to reach its limit for local resources, it can secure additional MAC addresses by using global resources. Global resources are shared among all the interfaces on a first-come, first-served basis.

The maximum number of MAC addresses any single interface can secure is 64 (the maximum number of local resources available to the interface), plus the number of global resources not allocated to other interfaces.

Configuring the MAC port security feature

To configure the MAC port security feature, you perform the following tasks:

- Enable the MAC port security feature
- Set the maximum number of secure MAC addresses for an interface
- Set the port security age timer
- Specify secure MAC addresses
- Configure the device to automatically save secure MAC addresses to the startup-config file
- Specify the action taken when a security violation occurs
- Deny specific MAC addresses
- Port Security MAC Violation Limits

Enabling the MAC port security feature

By default, the MAC port security feature is disabled on all interfaces. You can enable or disable the feature globally on all interfaces or on an individual interface.

To enable the feature globally, first go to the level for global port security and then enter **enable**, as follows.

```
NetIron(config)# global-port-security
NetIron(config-global-port-security)# enable
```

To disable the feature on all interfaces at once, do the following.

```
NetIron(config)# global-port-security
NetIron(config-global-port-security)#disable
```

Syntax: global-port-security

This command is for global enable port security.

To enable port security on a specific interface, first go to the level of a specific interface and then security level.

```
NetIron(config)# interface ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-port-security-e100-7/11)# enable
```

Syntax: enable

This command applies to a specific interface or global configuration. The interface level take precedence over the global configuration.

Syntax: disable

This command applies to a specific interface or global configuration. The interface level take precedence over the global configuration.

Setting the maximum number of secure MAC addresses for an interface

When the port security feature is enabled, the interface can store 1 secure MAC address. You can increase the number of MAC addresses that can be secured to a maximum of 64, plus the total number of global resources available.

For example, to configure interface 7/11 to have a maximum of 10 secure MAC addresses.

```
NetIron(config)# interface ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-if-e100-7/11)# maximum 10
```

Syntax: maximum <*number-of-addresses*>

The <*number-of-addresses*> parameter can be set to a number from 0 – (64 + the total number of global resources available) The total number of global resources is 4096. Setting the parameter to 0 prevents any addresses from being learned. The default is 1.

Setting the port security age timer

By default, a learned MAC address stays secure indefinitely. You can configure the device to age out secure MAC addresses after a specified amount of time and can do so for all timers globally of for a specific interface.

To set the port security age timer to 10 minutes on all interfaces, first go to the level for global security.

```
NetIron(config)# global-port-security
NetIron(config-global-port-security)# age 10
```

Syntax: global-port-security

Syntax: [no] age <*minutes*>

The default is 0 (never age out secure MAC addresses).

To set the port security age timer to 10 minutes on a specific interface, go to the interface level and then the port security level for that interface.

```
NetIron(config)# interface ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-port-security-e100-7/11)# age 10
```

Syntax: port security

Syntax: [no] age <*minutes*>

The default is 0 (never age out secure MAC addresses).

Specifying secure MAC addresses

To specify a secure MAC address on an interface, enter commands such as the following.

```
NetIron(config)# interface ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-port-security-e100-7/11)# secure 0050.DA18.747C
```

Syntax: [no] secure <*mac-address*>

Autosaving secure MAC addresses to the startup-config file

The learned MAC addresses can automatically be saved to the startup-config file at specified intervals. You can specifiy the autosave interval at the global level or for a specific interface. For example, to set a 20-minute autosave interval globally for learned secure MAC addresses on the router, enter the following commands.

```
NetIron(config)# global-port-security
NetIron(config-port-security)# autosave 20
```

Syntax: global-port-security

Syntax: [no] autosave <*minutes*>

The interval range is 15 – 1440 minutes. By default, secure MAC addresses are not autosaved to the startup-config file. To remove autosave intervals, use the **no** form of the **autosave** command.

To specifiy the interval or MAC address autosave on a port, first go to the level of that port and then the port security level for that port, as follows.

```
NetIron(config)# interface ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-port-security-e100-7/11)#autosave 20
```

Syntax: port security

Syntax: [no] autosave <*minutes*>

The interval range is 15 – 1440 minutes. By default, secure MAC addresses are not autosaved to the startup-config file. To remove autosave intervals, use the **no** form of the **autosave** command.

Specifying the action taken when a security violation occurs

A security violation can occur when a user tries to plug into a port where a MAC address is already locked, or the maximum number of secure MAC addresses has been exceeded. When a security violation occurs, an SNMP trap and Syslog message are generated.

In addition, you configure the device to take one of two actions when a security violation occurs: either drop packets from the violating address (and allow packets from secure addresses), or disable the port altogether for a specified amount of time.

To configure the device to drop packets from a violating address and allow packets from secure addresses.

```
NetIron(config)# interface ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-port-security-e100-7/11)# violation restrict
```

Syntax: violation restrict

To shut down the port for 5 minutes when a security violation occurs.

```
NetIron(config)# interface ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-port-security-e100-7/11)# violation shutdown 5
```

Syntax: violation shutdown <*minutes*>

To specific the mac-addresses that will be denied. All other mac-addresses no specificed will be allowed.

```
NetIron(config)# interface ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-port-security-e100-7/11)# deny-mac-address
```

Syntax: deny-mac-address

NOTE
When using this feature with a 24-port 10/100 module (part number B24E) only the **shutdown** option is supported. The **restrict** option is not supported on the B24E.

Denying specific MAC addresses

You can configure the *violation deny mode*. The violation deny mode allows you to deny MAC addresses on a global level or on a per port level.

Denying MAC addresses globally

To deny a specific MAC address globally, enable the violation deny mode, then specify the MAC address to be denied.

```
NetIron(config)# global-port-security
NetIron(config-port-security)# violation deny
NetIron(config-port-security)# deny-mac-address 0000.0000.0001 2
```

Global denied secure MAC addresses are denied system-wide. These MAC entries are added to the MAC table as deny entries, when a flow is received and are the only MAC addresses that are denied. All other MAC addresses are allowed.

A maximum of 512 deny MAC addresses can be configured on a global level.

Denying MAC addresses on an interface

You can specify which MAC addresses can be denied on an interface.

```
NetIron(config)# internet ethernet 7/11
NetIron(config-if-e100-7/11)# port security
NetIron(config-port-security-e100-7/11)# violation deny
NetIron(config-port-security-e100-7/11)# deny-mac-addr 0000.1111.2222 4
```

Only the configured MAC addresses are denied on the specified interface. All other MAC addresses are allowed.

A maximum of 64 deny MAC addresses can be configured at an interface level.

Displaying MAC addresses that have been denied

Use the **show port security global-deny** command to display all the MAC addresses that have been denied globally. Use the **show port security denied-macs** command to display all the denied MAC addresses

Port security MAC violation limit

Use the **violation restrict** command to specify how many packets the system can receive in a one-second interval from denied MAC address before the system shuts the port down.

Configuring port security

To enable this new mode, enter a command such as the following.

```
NetIron(config)# global-port-security
NetIron(config-port-security)# violation restrict 12
```

Syntax: violation restrict [#-denied-packets processed]

Enter 0 – 64000. This parameter has no default.

NOTE
With the introduction of this command, packets from denied MAC addresses are now processed in software by the LP. They are no longer programmed in the hardware.

In addition to the new processing of packets from denied MAC addresses, these packets can now be logged in the Syslog. And to prevent the Syslog from being overwhelmed with messages for denied packets, you can specify how many messages will be logged per second, based on a packet's IP address.

```
NetIron(config)# global-port-security
NetIron(config-port-security)# violation restrict 12
NetIron(config-port-security)# deny-log-rate <7>
```

Syntax: deny-log-rate [<#-*logs*>]

The #-logs parameter specifies the count per line card. Enter 1 – 10. There is no default.

The logged message contains the packet's IP address and the MAC address of the denied packet. For example, the following configuration shows that violation restrict is configured;

```
interface ethernet 14/1
 port security
  enable
  maximum 5
  violation restrict 1000
  secure-mac-address 0000.0022.2222 10
  secure-mac-address 0000.0022.2223 10
  secure-mac-address 0000.0022.2224 10
  secure-mac-address 0000.0022.2225 10
  secure-mac-address 0000.0022.2226 10
```

When packet from MAC address 000.0022.2227, an address that is not a secured MAC address, the following Syslog message is generated.

```
SYSLOG: Mar 10 17:36:12:<12>3-RW-Core-3, Interface e14/1 shutdn due to high rate
of denied mac 0000.0022.2227, vlan 10
SYSLOG: Mar 10 17:36:12:<14>3-RW-Core-3, Interface ethernet14/1, state
down - disabled
```

However, when **deny-log-rate** is configured,

```
interface ethernet 14/1
 disable
 port security
  enable
  maximum 5
  violation restrict 1000
  deny-log-rate 4
  secure-mac-address 0000.0022.2222 10
  secure-mac-address 0000.0022.2223 10
  secure-mac-address 0000.0022.2224 10
  secure-mac-address 0000.0022.2225 10
  secure-mac-address 0000.0022.2226 10
```

The following Syslog messages are generated.

```
Mar 10 17:38:51:I:Port security denied pkt: 0000.0022.2224 -> 0000.0011.1111
198.19.1.2 -> 198.19.1.1 [Protocol:114]
Mar 10 17:38:51:I:Port security denied pkt: 0000.0022.2224 -> 0000.0011.1111
198.19.1.2 -> 198.19.1.1 [Protocol:114]
Mar 10 17:38:51:I:Port security denied pkt: 0000.0022.2224 -> 0000.0011.1111
198.19.1.2 -> 198.19.1.1 [Protocol:114]
Mar 10 17:38:51:I:Port security denied pkt: 0000.0022.2224 -> 0000.0011.1111
198.19.1.2 -> 198.19.1.1 [Protocol:114]
Mar 10 17:38:51:I:Port security denied pkt: 0000.0022.2224 -> 0000.0011.1111
198.19.1.2 -> 198.19.1.1 [Protocol:114]
```

Displaying port security information

You can display the following information about the port security feature:

- The secure MAC addresses that have been saved to the startup-config file by the autosave feature
- The port security settings for an individual port or for all the ports on a specified module
- The secure MAC addresses configured on the device
- Port security statistics for an interface or for a module

Displaying port security settings

You can display the port security settings for an individual port or for all the ports on a specified module. For example, to display the port security settings for port 7/11, enter the following command.

```
NetIron# show port security e 1/1
```

```
Port  Security   MacAddrs          Violation  PortShutdn(minutes) SecureMac Learn
                Learnt/Max  Total/Count/Type   Status/Time/Remain   AgeTime
----- -------- ----------- ------------------ ------------------ ---------
------
 1/1  disabled    0/1         0/   0/shutdown    no/permanent         permanent
yes
```

Syntax: show port security <*module*> | <*portnum*>

This command displays the following information

TABLE 393 Output from the show port security <*module*> command

This field...	Displays...
Port	The slot and port number of the interface.
Security	Whether the port security feature has been enabled on the interface.
MacAddrs Learnt or Max	Learnt – The number of secure MAC addresses that have been learned on the interface. Max – The maximum number of secure MAC addresses that can be learned on the interface.
Violation Total or Count or Type	Total – The total number of violations that have occurred on the interface. Count – The count of the current violation on the interface. Type – The action to be undertaken when a security violation occurs, either “shutdown” or “restrict”.
PortShutdn (minutes) Status or Time or Remain	Status – Whether the interface has been shut down due to a security violation. Time – The number of seconds a port is shut down following a security violation, if the port is set to “shutdown” when a violation occurs. Remain – The number of seconds before the port is enabled again.
SecureMac AgeTime	The amount of time, in minutes, MAC addresses learned on the port will remain secure.
Learn	Whether the port is able to learn MAC addresses.

Displaying the secure MAC addresses on the device

To list the secure MAC addresses configured on the device, enter the following command.

```
NetIron(config)# show port security mac
Port  Num-Addr Secure-Src-Addr Resource Age-Left  Shutdown/Time-Left
----- -------- -------------- -------- --------- ------------------
 7/11        1  0050.da18.747c    Local        10       no
```

Syntax: show port security mac

This command displays the following information.

TABLE 394 Output from the show port security mac command

This field...	Displays...
Port	The slot and port number of the interface.
Num-Addr	The number of MAC addresses secured on this interface.
Secure-Src-Addr	The secure MAC address.
Resource	Whether the address was secured using a local or global resource. Refer to “Local and global resources” on page 2052 for more information.

TABLE 394 Output from the show port security mac command (Continued)

This field...	Displays...
Age-Left	The number of minutes the MAC address will remain secure.
Shutdown or Time-Left	Whether the interface has been shut down due to a security violation and the number of seconds before it is enabled again.

Displaying port security statistics

You can display port security statistics for an interface or for a module.

For example, to display port security statistics for interface 7/11.

```
NetIron# show port security statistics e 7/11
Port  Total-Addrs Maximum-Addrs Violation Shutdown/Time-Left
----- ----------- ------------- --------- ------------------
 7/11           1             1         0       no
```

Syntax: show port security statistics <*portnum*>

TABLE 395 Output from the show port security statistics <*portnum*> command

This field...	Displays...
Port	The slot and port number of the interface.
Total-Addrs	The total number of secure MAC addresses on the interface.
Maximum-Addrs	The maximum number of secure MAC addresses on the interface.
Violation	The number of security violations on the port.
Shutdown or Time-Left	Whether the port has been shut down due to a security violation and the number of seconds before it is enabled again.

To display port security statistics for a module, enter the following command.

```
NetIron# show port security statistics 7
Module 7:
  Total ports: 0
  Total MAC address(es): 0
  Total violations: 0
  Total shutdown ports 0
```

Syntax: show port security statistics <*module*>

TABLE 396 Output from the show port security statistics <*module*> command

This field...	Displays...
Total ports:	The number of ports on the module.
Total MAC address(es):	The total number of secure MAC addresses on the module.
Total violations:	The number of security violations encountered on the module.
Total shutdown ports:	The number of ports on the module shut down as a result of security violations.

Chapter

Configuring 802.1x Port Security

51

PowerConnect B-MLXe supports the following 802.1x Port Security features:

- 802.1x Port Security
- 802.1x Port Security and sFlow
- Dynamic VLAN Assignment for 802.1x Ports
- Strict Security Mode for Dynamic Filter Assignment
- Dynamically Applying Existing ACLs or MAC Address Filter
- Per-User IP ACLs or MAC Address Filters
- Periodic Re-Authentication
- Re-Authenticating a Port Manually
- Quiet Periods
- EAP-Request or Identity Frame Retransmissions
- Timeouts for Retransmission of Messages to the Authentication Server
- Timeout for Retransmission of EAP-Request Frames to the client
- Allowing Multiple 802.1x clients to Authenticate

Overview of 802.1x port security

The Multi-Service IronWare software supports the IEEE 802.1x standard for authenticating devices attached to LAN ports. Using 802.1x port security, you can configure a device to grant access to a port based on information supplied by a client to an authentication server.

When a user logs on to a network that uses 802.1x port security, the device grants (or does not grant) access to network services after the user is authenticated by an authentication server. The user-based authentication in 802.1x port security provides an alternative to granting network access based on a user's IP address, MAC address, or subnetwork.

IETF RFC support

The implementation of 802.1x port security supports the following RFCs:

- RFC 2284 PPP Extensible Authentication Protocol (EAP)
- RFC 2865 Remote Authentication Dial In User Service (RADIUS)
- RFC 2869 RADIUS Extensions

How 802.1x port security works

This section explains the basic concepts behind 802.1x port security, including device roles, how the devices communicate, and the procedure used for authenticating clients.

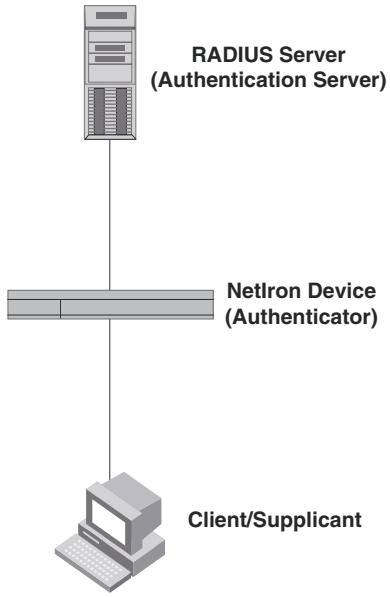
Device roles in an 802.1x configuration

The 802.1x standard defines the roles of **client or Supplicant**, **Authenticator**, and **Authentication Server** in a network.

The client (known as a **Supplicant** in the 802.1x standard) provides username or password information to the Authenticator. The Authenticator sends this information to the Authentication Server. Based on the client's information, the Authentication Server determines whether the client can use services provided by the Authenticator. The Authentication Server passes this information to the Authenticator, which then provides services to the client, based on the authentication result.

Figure 224 illustrates these roles.

FIGURE 224 Authenticator, client or supplicant, and authentication server in an 802.1x configuration



Authenticator – The device that controls access to the network. In an 802.1x configuration, the device serves as the Authenticator. The Authenticator passes messages between the client and the Authentication Server. Based on the identity information supplied by the client, and the authentication information supplied by the Authentication Server, the Authenticator either grants or does not grant network access to the client.

client or supplicant – The device that seeks to gain access to the network. clients must be running software that supports the 802.1x standard (for example, the Windows XP operating system). clients can either be directly connected to a port on the Authenticator, or can be connected by way of a hub.

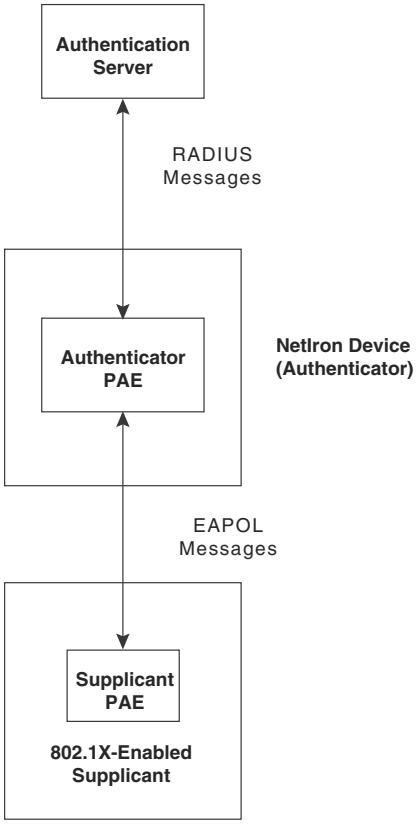
Authentication server – The device that validates the client and specifies whether or not the client may access services on the device. The device supports Authentication Servers running RADIUS.

Communication between the devices

For communication between the devices, 802.1x port security uses the Extensible Authentication Protocol (AP), defined in RFC 2284. The 802.1x standard specifies a method for encapsulating EAP messages so that they can be carried over a LAN. This encapsulated form of EAP is known as EAP over LAN (EAPOL). The standard also specifies a means of transferring the EAPOL information between the client or Supplicant, Authenticator, and Authentication Server.

EAPOL messages are passed between the Port Access Entity (PAE) on the Supplicant and the Authenticator. Figure 225 shows the relationship between the Authenticator PAE and the Supplicant PAE.

FIGURE 225 Authenticator PAE and supplicant PAE



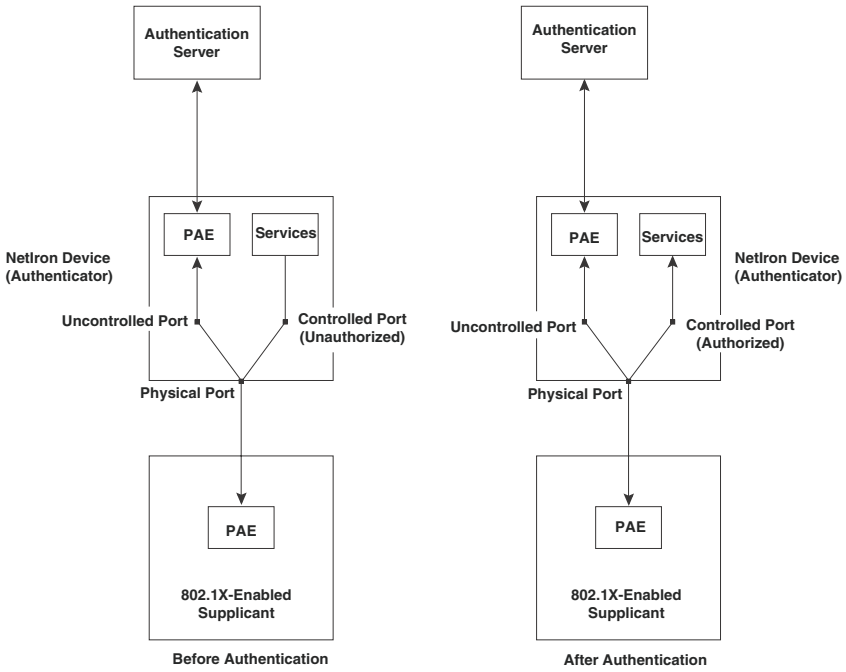
Authenticator PAE – The Authenticator PAE communicates with the Supplicant PAE, receiving identifying information from the Supplicant. Acting as a RADIUS client, the Authenticator PAE passes the Supplicant's information to the Authentication Server, which decides whether the Supplicant can gain access to the port. If the Supplicant passes authentication, the Authenticator PAE grants it access to the port.

Supplicant PAE – The Supplicant PAE supplies information about the client to the Authenticator PAE and responds to requests from the Authenticator PAE. The Supplicant PAE can also initiate the authentication procedure with the Authenticator PAE, as well as send logoff messages.

Controlled and uncontrolled ports

A physical port on the device used with 802.1x port security has two virtual access points: a controlled port and an uncontrolled port. The controlled port provides full access to the network. The uncontrolled port provides access only for EAPOL traffic between the client and the Authentication Server. When a client is successfully authenticated, the controlled port is opened to the client. Figure 226 illustrates this concept.

FIGURE 226 Controlled and uncontrolled ports before and after client authentication



Before a client is authenticated, only the uncontrolled port on the Authenticator is open. The uncontrolled port allows only EAPOL frames to be exchanged between the client and the Authentication Server. The controlled port is in the unauthorized state and allows no traffic to pass through.

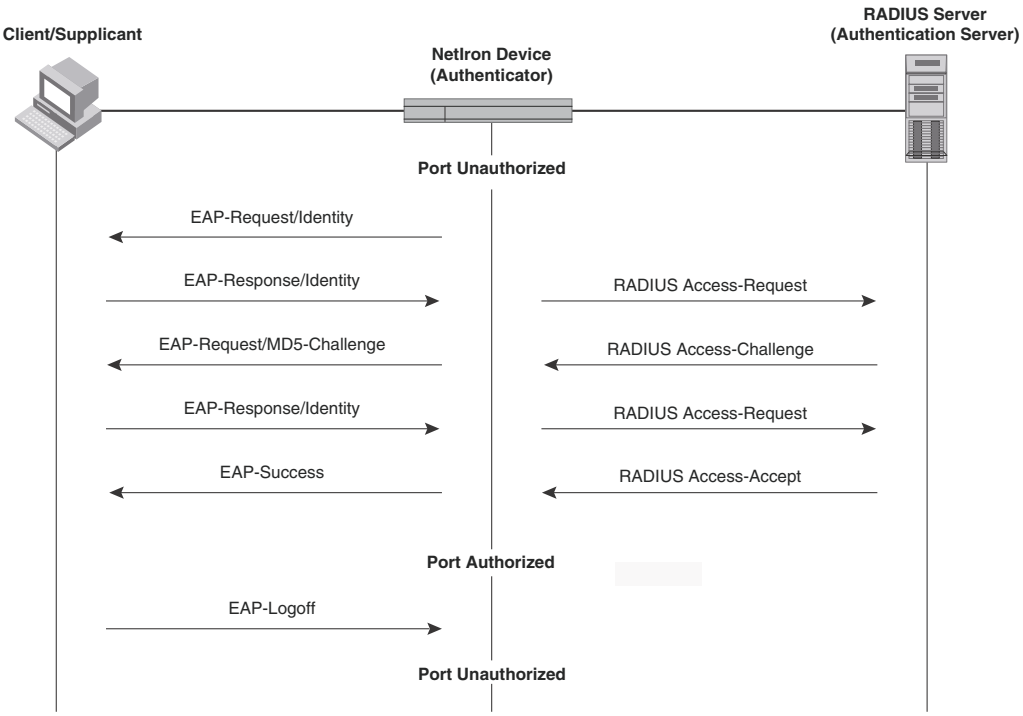
During authentication, EAPOL messages are exchanged between the Supplicant PAE and the Authenticator PAE, and RADIUS messages are exchanged between the Authenticator PAE and the Authentication Server. Refer to "Message exchange during authentication" on page 2065 for an example of this process. If the client is successfully authenticated, the controlled port becomes authorized, and traffic from the client can flow through the port normally.

By default, all controlled ports on the device are placed in the authorized state, allowing all traffic. When authentication is activated on an 802.1x-enabled interface, the controlled port on the interface is placed initially in the unauthorized state. When a client connected to the port is successfully authenticated, the controlled port is then placed in the authorized state until the client logs off. Refer to "Enabling 802.1x port security" on page 2074 for more information.

Message exchange during authentication

Figure 227 illustrates a sample exchange of messages between an 802.1x-enabled client, a device acting as Authenticator, and a RADIUS server acting as an Authentication Server.

FIGURE 227 Message exchange during authentication



In this example, the Authenticator (the device) initiates communication with an 802.1x-enabled client. When the client responds, it is prompted for a username (255 characters maximum) and password. The Authenticator passes this information to the Authentication Server, which determines whether the client can access services provided by the Authenticator. When the client is successfully authenticated by the RADIUS server, the port is authorized. When the client logs off, the port becomes unauthorized again.

Dell's 802.1x implementation supports dynamic VLAN assignment. If one of the attributes in the Access-Accept message sent by the RADIUS server specifies a VLAN identifier, and this VLAN is available on the device, the client's port is moved from its default VLAN to the specified VLAN. When the client disconnects from the network, the port is placed back in its default VLAN. Refer to "Configuring dynamic VLAN assignment for 802.1x ports" on page 2070 for more information.

Dell's 802.1x implementation supports dynamically applying an IP ACL or MAC address filter to a port, based on information received from the Authentication Server.

If a client does not support 802.1x, authentication cannot take place. The device sends EAP-Request or Identity frames to the client, but the client does not respond to them.

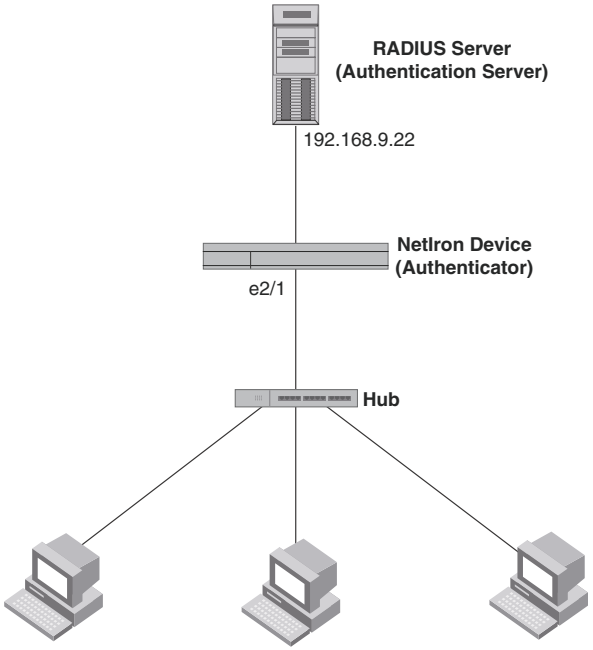
When a client that supports 802.1x attempts to gain access through a non-802.1x-enabled port, it sends an EAP start frame to the device. When the device does not respond, the client considers the port to be authorized, and starts sending normal traffic.

Dell routers and switches support MD5-challenge TLS and any other EAP-encapsulated authentication types in EAP Request or Response messages. In other words, the devices are transparent to the authentication scheme used.

Authenticating multiple clients connected to the same port

Dell routers and switches support 802.1x authentication for ports with more than one client connected to them. Figure 228 illustrates a sample configuration where multiple clients are connected to a single 802.1x port.

FIGURE 228 Multiple clients connected to a single 802.1x-enabled port



If there are multiple clients connected to a single 802.1x-enabled port, the device authenticates each of them individually. Each client's authentication status is independent of the others, so that if one authenticated client disconnects from the network, it has no effect on the authentication status of any of the other authenticated clients.

By default, traffic from clients that cannot be authenticated by the RADIUS server is dropped in hardware. You can optionally configure the device to assign the port to a "restricted" VLAN if authentication of the client is unsuccessful.

How 802.1x multiple client authentication works

When multiple clients are connected to a single 802.1x-enabled port on a router (as in Figure 228), 802.1x authentication is performed in the following ways.

1. One of the 802.1x-enabled clients attempts to log into a network in which a device serves as an Authenticator.
2. The device creates an internal session (called a ***dot1x-mac-session***) for the client. A dot1x-mac-session serves to associate a client's MAC address and username with its authentication status.
3. The device performs 802.1x authentication for the client. Messages are exchanged between the device and the client, and between the device and the Authentication Server (RADIUS server). The result of this process is that the client is either successfully authenticated or not authenticated, based on the username and password supplied by the client.
4. If the client is successfully authenticated, the client's dot1x-mac-session is set to "access-is-allowed". This means that traffic from the client can be forwarded normally.
5. If authentication for the client is unsuccessful the first time, multiple attempts to authenticate the client will be made as determined by the **attempts** variable in the **auth-fail-max-attempts** command.
    - Refer to "Specifying the number of authentication attempts the device makes before dropping packets" on page 2080 for information on how to do this.
6. If authentication for the client is unsuccessful more than the number of times specified by the **attempts** variable in the **auth-fail-max-attempts** command, an ***authentication-failure*** action is taken. The authentication-failure action can be either to drop traffic from the client, or to place the port in a "restricted" VLAN:
    - If the authentication-failure action is to drop traffic from the client, then the client's dot1x-mac-session is set to "access-denied", causing traffic from the client to be dropped in hardware.
    - If the authentication-failure action is to place the port in a "restricted" VLAN, If the client's dot1x-mac-session is set to "access-restricted" then the port is moved to the specified restricted VLAN, and traffic from the client is forwarded normally.
7. When the client disconnects from the network, the device deletes the client's dot1x-mac-session. This does not affect the dot1x-mac-session or authentication status (if any) of the other clients connected on the port.

NOTES:

- The client's dot1x-mac-session establishes a relationship between the username and MAC address used for authentication. If a user attempts to gain access from different clients (with different MAC addresses), he or she would need to be authenticated from each client.
- If a client has been denied access to the network (that is, the client's dot1x-mac-session is set to "access-denied"), then you can cause the client to be re-authenticated by manually disconnecting the client from the network, or by using the **clear dot1x mac-session** command. Refer to "Clearing a dot1x-mac-session for a MAC address" on page 2080 for information on this command.
- When a client has been denied access to the network, its dot1x-mac-session is aged out if no traffic is received from the client's MAC address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. You can optionally change the software aging period for dot1x-mac-sessions or disable aging altogether. After the denied client's dot1x-mac-session is aged out, traffic from that client is no longer blocked, and the client can be re-authenticated.

802.1x port security and sFlow

sFlow is a system for observing traffic flow patterns and quantities within and among a set of the devices. sFlow works by taking periodic samples of network data and exporting this information to a collector.

When you enable sFlow forwarding on an 802.1x-enabled interface, the samples taken from the interface include the user name string at the inbound or outbound port, if that information is available.

For more information on sFlow, refer to Chapter 57, “sFlow”.

Configuring 802.1x port security

Configuring 802.1x port security on a device consists of the following tasks.

1. Configuring device interaction with the Authentication Server:
   - “Configuring an authentication method list for 802.1x” on page 2069
   - “Setting RADIUS parameters” on page 2069
   - “Configuring dynamic VLAN assignment for 802.1x ports” on page 2070 (optional)
2. Configuring the device role as the Authenticator:
   - “Enabling 802.1x port security” on page 2074
   - “Initializing 802.1x on a port” on page 2079 (optional)
3. Configuring device interaction with clients:
   - “Configuring periodic re-authentication” on page 2076 (optional)
   - “Re-authenticating a port manually” on page 2077 (optional)
   - “Setting the quiet period” on page 2077 (optional)
   - “Setting the interval for retransmission of EAP-request or identity frames” on page 2077 (optional)
   - “Specifying the number of EAP-request or identity frame retransmissions” on page 2078 (optional)
   - “Specifying a timeout for retransmission of EAP-request frames to the client” on page 2078 (optional)
   - “Allowing multiple 802.1x clients to authenticate” on page 2079 (optional)

NOTE
Multi-Device Port Authentication and 802.1x authentication can both be enabled on a port; however only one of them can authenticate a MAC address or 802.1x client. Refer to “Support for multi-device port authentication and 802.1x on the same interface” on page 2037.

Configuring an authentication method list for 802.1x

To use 802.1x port security, you must specify an authentication method to be used to authenticate clients. The device supports RADIUS authentication with 802.1x port security. To use RADIUS authentication with 802.1x port security, you create an authentication method list for 802.1x and specify RADIUS as an authentication method, then configure communication between the device and RADIUS server.

Example

```
NetIron(config)# aaa authentication dot1x default radius
```

Syntax: [**no**] **aaa authentication dot1x default** <*method-list*>

For the <*method-list*>, enter at least one of the following authentication methods:

radius – Use the list of all RADIUS servers that support 802.1x for authentication.

none – Use no authentication. The client is automatically authenticated without the device using information supplied by the client.

NOTE
If you specify both **radius** and **none**, make sure **radius** comes before **none** in the method list.

Setting RADIUS parameters

To use a RADIUS server to authenticate access to a device, you must identify the server to the device.

```
NetIron(config)# radius-server host 209.157.22.99 auth-port 1812 acct-port 1813
default key mirabeau dot1x
```

Syntax: **radius-server host** <*ip-addr*> | <*server-name*> [**auth-port** <*number*> **acct-port** <*number*> [**authentication-only** | **accounting-only** | **default** [**key 0** | 1 <*string*> [**dot1x**]]]]

The **host** <*ip-addr*> | <*server-name*> parameter is either an IP address or an ASCII text string.

The **auth-port** <*number*> parameter specifies what port to use for RADIUS authentication.

The **acct-port** <*number*> parameter specifies what port to use for RADIUS accounting.

The **dot1x** parameter indicates that this RADIUS server supports the 802.1x standard. A RADIUS server that supports the 802.1x standard can also be used to authenticate non-802.1x authentication requests.

NOTE
To implement 802.1x port security, at least one of the RADIUS servers identified to the device must support the 802.1x standard.

Supported RADIUS attributes

Many IEEE 802.1x Authenticators will function as RADIUS clients. Some of the RADIUS attributes may be received as part of IEEE 802.1x authentication. The device supports the following RADIUS attributes for IEEE 802.1x authentication:

- Username (1) – RFC 2865
- FilterId (11) – RFC 2865
- Vendor-Specific Attributes (26) – RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 2579
- Tunnel-Private-Group-Id (81) – RFC 2868

Configuring dynamic VLAN assignment for 802.1x ports

Dell's 802.1x implementation supports assigning a port to a VLAN dynamically, based on information received from an Authentication (RADIUS) Server. If one of the attributes in the Access-Accept message sent by the RADIUS server specifies a VLAN identifier, and this VLAN matches a VLAN on the device, the client's port is moved from its default VLAN to the specified VLAN. When the client disconnects from the network, the port is placed back in its default VLAN.

When a client or supplicant successfully completes the EAP authentication process, the Authentication Server (the RADIUS server) sends the Authenticator (the device) a RADIUS Access-Accept message that grants the client access to the network. The RADIUS Access-Accept message contains attributes set for the user in the user's access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies a VLAN identifier, and this VLAN is available on the device, the client's port is moved from its default VLAN to the specified VLAN. When the client disconnects from the network, the port is placed back in its default VLAN.

NOTE
This feature is supported on port-based VLANs only. This feature cannot be used to place an 802.1x-enabled port into a Layer 3 protocol VLAN.

To enable 802.1x VLAN ID support on the device, you must add the following attributes to a user's profile on the RADIUS server.

TABLE 397 802.1x VLAN attributes required from the RADIUS server

Attribute name	Type	Value
Tunnel-Type	064	13 (decimal) – VLAN
Tunnel-Medium-Type	065	6 (decimal) – 802
Tunnel-Private-Group-ID	081	<*vlan-name*> (string) – either the name or the number of a VLAN configured on the device.

The device reads the attributes as follows:

- If the Tunnel-Type or the Tunnel-Medium-Type attributes in the Access-Accept message do not have the values specified above, the device ignores the three Attribute-Value pairs. The client becomes authorized, but the client's port is not dynamically placed in a VLAN.
- If the Tunnel-Type or the Tunnel-Medium-Type attributes in the Access-Accept message do have the values specified above, but there is no value specified for the Tunnel-Private-Group-ID attribute, the client will not become authorized.

- When the device receives the value specified for the Tunnel-Private-Group-ID attribute, it checks whether the <*vlan-name*> string matches the name of a VLAN configured on the device. If there is a VLAN on the device whose name matches the <*vlan-name*>, then the client's port is placed in the VLAN whose ID corresponds to the VLAN name.
- If the <*vlan-name*> string does not match the name of a VLAN, the device checks whether the string, when converted to a number, matches the ID of a VLAN configured on the device. If it does, then the client's port is placed in the VLAN with that ID.
- If the <*vlan-name*> string does not match either the name or the ID of a VLAN configured on the device, then the client will not become authorized.

The **show interface** command displays the VLAN to which an 802.1x-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port's default VLAN). Refer to "Displaying dynamically assigned VLAN information" on page 2084 for sample output indicating the port's dynamically assigned VLAN.

Considerations for dynamic VLAN assignment in an 802.1x multiple client configuration

The following considerations apply when a client in a 802.1x multiple client configuration is successfully authenticated, and the RADIUS Access-Accept message specifies a VLAN for the port:

- If the port is not already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of a valid VLAN on the device, then the port is placed in that VLAN.
- If the port is already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of a different VLAN, then it is considered an authentication failure. The port's VLAN membership is not changed.
- If the port is already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of that same VLAN, then traffic from the client is forwarded normally.
- If the RADIUS Access-Accept message specifies the name or ID of a VLAN that does not exist on the device, then it is considered an authentication failure.
- If the RADIUS Access-Accept message does not contain any VLAN information, the client's dot1x-mac-session is set to "access-is-allowed". If the port is already in a RADIUS-specified VLAN, it remains in that VLAN.

Disabling and enabling strict security mode for dynamic filter assignment

By default, 802.1x dynamic filter assignment operates in strict security mode. When strict security mode is enabled, 802.1x authentication for a port fails if the Filter-ID attribute contains invalid information, or if insufficient system resources are available to implement the per-user IP ACLs or MAC address filters specified in the Vendor-Specific attribute.

When strict security mode is enabled:

- If the Filter-ID attribute in the Access-Accept message contains a value that does not refer to an existing filter (that is, a MAC address filter or IP ACL configured on the device), then the client will not be authenticated, regardless of any other information in the message (for example, if the Tunnel-Private-Group-ID attribute specifies a VLAN to which to assign the port).

- If the Vendor-Specific attribute specifies the syntax for a filter, but there are insufficient system resources to implement the filter, then the port will not be authenticated.
- If the device does not have the system resources available to dynamically apply a filter to a port, then the port will not be authenticated.

NOTE
If the Access-Accept message contains values for both the Filter-ID and Vendor-Specific attributes, then the value in the Vendor-Specific attribute (the per-user filter) takes precedence.
Also, if authentication for a port fails because the Filter-ID attribute referred to a non-existent filter, or there were insufficient system resources to implement the filter, then a Syslog message is generated.

When strict security mode is disabled:

- If the Filter-ID attribute in the Access-Accept message contains a value that does not refer to an existing filter (that is, a MAC address filter or IP ACL configured on the device), then the port is still authenticated, but no filter is dynamically applied to it.
- If the Vendor-Specific attribute specifies the syntax for a filter, but there are insufficient system resources to implement the filter, then the port is still authenticated, but the filter specified in the Vendor-Specific attribute is not applied to the port.

By default, strict security mode is enabled for all 802.1x-enabled interfaces, but you can manually disable or enable it, either globally or for specific interfaces.

To disable strict security mode globally, enter the following commands.

```
NetIron(config)# dot1x-enable
NetIron(config-dot1x)# no global-filter-strict-security
```

After you have globally disabled strict security mode on the device, you can re-enable it by entering the following command.

```
NetIron(config-dot1x)# global-filter-strict-security
```

Syntax: [no] **global-filter-strict-security**

To disable strict security mode for a specific interface, enter commands such as the following.

```
NetIron(config)# interface e 1
NetIron(config-if-e10000-1)# no dot1x filter-strict-security
```

To re-enable strict security mode for an interface, enter the following command.

```
NetIron(config-if-e10000-1)# dot1x filter-strict-security
```

Syntax: [no] **dot1x filter-strict-security**

The output of the **show dot1x** and **show dot1x config** commands has been enhanced to indicate whether strict security mode is enabled or disabled globally and on an interface.

Dynamically applying existing ACLs or MAC address filter

When a port is authenticated using 802.1x security, an IP ACL or MAC address filter that exists in the running configuration on the device can be dynamically applied to the port. To do this, you configure the Filter-ID (type 11) attribute on the RADIUS server. The Filter-ID attribute specifies the name or number of the IP ACL or MAC address filter.

The following table shows the syntax for configuring the Filter-ID attribute to refer to a IP ACL or MAC address filter.

Value	Description
ip.<*number*>.in	Applies the specified numbered ACL to the 802.1x authenticated port in the inbound direction.
ip.<*name*>.in	Applies the specified named ACL to the 802.1x authenticated port in the inbound direction.
ip.<*number*>.out	Applies the specified numbered ACL to the 802.1x authenticated port in the outbound direction.
ip.<*name*>.out	Applies the specified named ACL to the 802.1x authenticated port in the outbound direction.
mac.<*number*>.in	Applies the specified MAC address filter to the 802.1x authenticated port in the inbound direction.

The following table lists examples of values you can assign to the Filter-ID attribute on the RADIUS server to refer to IP ACLs and MAC address filters configured on a device.

Possible values for the filter ID attribute on the RADIUS server	ACL or MAC address filter configured on the device
ip.2.in	access-list 2 permit host 36.48.0.3 access-list 2 permit 36.0.0.0 0.255.255.255
ip.102.in	access-list 102 permit ip 36.0.0.0 0.255.255.255 any
ip.fdry_filter.in	ip access-list standard fdry_filter permit host 36.48.0.3
mac.401.in	access-list 401 permit 3333.3333.3333 ffff.ffff.ffff any etype any
mac.402.in mac.403.in	access-list 402 permit 3333.3333.3333 ffff.ffff.ffff any etype any access-list 403 permit 2222.2222.2222 ffff.ffff.ffff any etype any

NOTES:

- The <*name*> in the Filter ID attribute is case-sensitive.
- You can specify only numbered MAC address filters in the Filter ID attribute. Named MAC address filters are not supported.
- Dynamic ACL filters are supported for the inbound and outbound direction.
- MAC address filters are supported only for the inbound direction. Outbound MAC address filters are not supported.
- Dynamically assigned IP ACLs and MAC address filters are subject to the same configuration restrictions as non-dynamically assigned IP ACLs and MAC address filters.

- Multiple IP ACLs and MAC address filters can be specified in the Filter ID attribute, allowing multiple address filters to be simultaneously applied to an 802.1x authenticated port. Use commas, semicolons, or carriage returns to separate the address filters (for example: ip.3.in,mac.402.in).
- If 802.1x is enabled on a VE port, ACLs, dynamic (802.1x assigned) or static (user configured), cannot be applied to the port.

Configuring per-user IP ACLs or MAC address filters

Per-user IP ACLs and MAC address filters make use of the Vendor-Specific (type 26) attribute to dynamically apply filters to ports. Defined in the Vendor-Specific attribute are Dell ACL or MAC address filter statements. When the RADIUS server returns the Access-Accept message granting a client access to the network, the device reads the statements in the Vendor-Specific attribute and applies these IP ACLs or MAC address filters to the client's port. When the client disconnects from the network, the dynamically applied filters are no longer applied to the port. If any filters had been applied to the port previous to the client connecting, then those filters are reapplied to the port.

The following is the syntax for configuring the PowerConnect Vendor-Specific attribute with ACL or MAC address filter statements.

Value	Description
ipacl.e.in=<*extended-acl-entries*>	Applies the specified extended ACL entries to the 802.1x authenticated port in the inbound direction.
ipacl.e.out=<*extended-acl-entries*>	Applies the specified extended ACL entries to the 802.1x authenticated port in the outbound direction.
macfilter.in=<*mac-accesslist-entries*>	Applies the specified MAC address filter entries to the 802.1x authenticated port in the inbound direction.

The following table shows examples of IP ACLs and MAC address filters configured in the Dell Vendor-Specific attribute on a RADIUS server. These IP ACLs and MAC address filters follow the same syntax as other Dell ACLs and MAC address filters. Refer to 21, "Access Control List" or Chapter 40, "Configuring an IPv6 Access Control List" for information on syntax.

IP ACL or MAC address filter	Vendor-specific attribute on RADIUS server
Extended ACL with one entry (outbound direction)	ipacl.e.out=permit ip 36.0.0.0 0.255.255.255 any
Mac address filter with one entry	macfilter.in= deny any any
Mac address filter with two entries	macfilter.in= permit 0000.0000.3333 ffff.ffff.0000 any, macfilter.in= permit 0000.0000.4444 ffff.ffff.0000 any

The RADIUS server allows one instance of the Vendor-Specific attribute to be sent in an Access-Accept message. However, the Vendor-Specific attribute can specify multiple IP ACLs or MAC address filters. You can use commas, semicolons, or carriage returns to separate the filters (for example: ipacl.e.in= permit ip any any,ipacl.e.in = deny ip any any).

Enabling 802.1x port security

By default, 802.1x port security is disabled on devices. To enable the feature on the device and enter the dot1x configuration level, enter the following command.

```
NetIron(config)# dot1x-enable
NetIron(config-dot1x)#
```

Syntax: [no] **dot1x-enable**

At the dot1x configuration level, you can enable 802.1x port security on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to enable 802.1x port security on all interfaces on the device, enter the following command.

```
NetIron(config-dot1x)# enable all
```

Syntax: [no] **enable all**

To enable 802.1x port security on interface 3/11, enter the following command.

```
NetIron(config-dot1x)# enable ethernet 3/11
```

Syntax: [no] **enable** <*portnum*>

To enable 802.1x port security on interfaces 3/11 through 3/16, enter the following command.

```
NetIron(config-dot1x)# enable ethernet 3/11 to 3/16
```

Syntax: [no] **enable** <*portnum*> **to** <*portnum*>

Setting the port control

To activate authentication on an 802.1x-enabled interface, you specify the kind of ***port control*** to be used on the interface. An interface used with 802.1x port security has two virtual access points:

- The controlled port can be either the authorized or unauthorized state. In the authorized state, it allows normal traffic to pass between the client and the authenticator. In the unauthorized state, it allows no traffic to pass through.
- The uncontrolled port allows only EAPOL traffic between the client and the authentication server.

Refer to Figure 226 on page 2064 for an illustration of this concept.

By default, all controlled ports on the device are in the authorized state, allowing all traffic. When you activate authentication on an 802.1x-enabled interface, its controlled port is placed in the unauthorized state. When a client connected to the interface is successfully authenticated, the controlled port is then placed in the authorized state for that client. The controlled port remains in the authorized state until the client logs off.

To activate authentication on an 802.1x-enabled interface, you configure the interface to place its controlled port in the authorized state when a client is authenticated by an authentication server. To do this, enter commands such as the following.

```
NetIron(config)# interface e 3/1
NetIron(config-if-e10000-3/1)# dot1x port-control auto
```

Syntax: [no] **dot1x port-control** [**force-authorized** | **force-unauthorized** | **auto**]

When an interface's control type is set to **auto**, its controlled port is initially set to unauthorized, but is changed to authorized when the connecting client is successfully authenticated by an Authentication Server.

The port control type can be one of the following:

force-authorized – The port's controlled port is placed unconditionally in the authorized state, allowing all traffic. This is the default state for ports on the device. Also, this parameter allows connection from multiple clients.

force-unauthorized – The controlled port is placed unconditionally in the unauthorized state.

auto – The controlled port is unauthorized until authentication takes place between the client and Authentication Server. Once the client passes authentication, the port becomes authorized. This has the effect of activating authentication on an 802.1x-enabled interface.

Notes

You cannot enable 802.1x port security on ports that have any of the following features enabled:

- Static MAC configurations
- Link aggregation
- Metro Ring Protocol (MRP)
- Tagged port
- Mirror port
- LAG port
- MAC port security
- Management Port
- VE members

Configuring periodic re-authentication

You can configure the device to periodically re-authenticate clients connected to 802.1x-enabled interfaces. When you enable periodic re-authentication, the device re-authenticates clients every 3,600 seconds by default. You can optionally specify a different re-authentication interval of between 1 – 4294967295 seconds.

To configure periodic re-authentication using the default interval of 3,600 seconds, enter the following command.

```
NetIron(config)#dot1x-enable
NetIron(config-dot1x)# re-authentication
```

Syntax: [**no**] **re-authentication**

To configure periodic re-authentication with an interval of 2,000 seconds, enter the following commands.

```
NetIron(config)#dot1x-enable
NetIron(config-dot1x)# re-authentication
NetIron(config-dot1x)# timeout re-authperiod 2000
```

Syntax: [**no**] **timeout re-authperiod** <*seconds*>

The re-authentication interval is a global setting, applicable to all 802.1x-enabled interfaces. If you want to re-authenticate clients connected to a specific port manually, use the **dot1x re-authenticate** command. Refer to "Re-authenticating a port manually".

Re-authenticating a port manually

When periodic re-authentication is enabled, by default the device re-authenticates clients connected to an 802.1x-enabled interface every 3,600 seconds (or the time specified by the **dot1x timeout re-authperiod** command). You can also manually re-authenticate clients connected to a specific port.

For example, to re-authenticate clients connected to interface 3/1, enter the following command.

```
NetIron# dot1x re-authenticate e 3/1
```

Syntax: [**no**] **dot1x re-authenticate** <*portnum*>

Setting the quiet period

If the device is unable to authenticate the client, the device waits a specified amount of time before trying again. The amount of time the device waits is specified with the **quiet-period** parameter. This timer also indicates how long a client that failed authentication would have its blocked entry programmed into the hardware.The **quiet-period** parameter can be from 0 – 4294967295 seconds. The default is 60 seconds.

For example, to set the quiet period to 30 seconds, enter the following command.

```
NetIron(config-dot1x)# timeout quiet-period 30
```

Syntax: [**no**] **timeout quiet-period** <*seconds*>

Setting the interval for retransmission of EAP-request or identity frames

When the device sends a client an EAP-request or identity frame, it expects to receive an EAP-response or identity frame from the client. If the client does not send back an EAP-response or identity frame, the device waits a specified amount of time and then retransmits the EAP-request or identity frame. You can specify the amount of time the device waits before retransmitting the EAP-request or identity frame to the client. This amount of time is specified with the **tx-period** parameter. The **tx-period** parameter can be from 1 – 65535 seconds. The default is 30 seconds.

For example, to cause the device to wait 60 seconds before retransmitting an EAP-request or identity frame to a client, enter the following command.

```
NetIron(config-dot1x)# timeout tx-period 60
```

Syntax: [**no**] **timeout tx-period** <*seconds*>

If the client does not send back an EAP-response or identity frame within 60 seconds, the device will transmit another EAP-request or identity frame.

Specifying the number of EAP-request or identity frame retransmissions

If the device does not receive a EAP-response or identity frame from a client, the device waits 30 seconds (or the amount of time specified with the **timeout tx-period** command), then retransmits the EAP-request or identity frame. By default, the device retransmits the EAP-request or identity frame a maximum of two times. If no EAP-response or identity frame is received from the client after two EAP-request or identity frame retransmissions, the device restarts the authentication process with the client.

You can optionally specify between 1 – 10 frame retransmissions. For example, to configure the device to retransmit an EAP-request or identity frame to a client a maximum of three times, enter the following command.

```
NetIron(config-dot1x)# maxreq 3
```

Syntax: maxreq <*value*>

Specifying a timeout for retransmission of messages to the Authentication Server

When performing authentication, the device receives EAPOL frames from the client and passes the messages on to the RADIUS server. The device expects a response from the RADIUS server within 30 seconds. If the RADIUS server does not send a response within 30 seconds, the device retransmits the message to the RADIUS server. The time constraint for retransmission of messages to the Authentication Server can be between 1 – 4294967295 seconds.

For the device, the possible values are: 1 - 4294967295.

For example, to configure the device to retransmit a message if the Authentication Server does not respond within 45 seconds, enter the following command.

```
NetIron(config-dot1x)# servertimeout 45
```

Syntax: servertimeout <*seconds*>

Specifying a timeout for retransmission of EAP-request frames to the client

Acting as an intermediary between the RADIUS Authentication Server and the client, the device receives RADIUS messages from the RADIUS server, encapsulates them as EAPOL frames, and sends them to the client. When the device relays an EAP-Request frame from the RADIUS server to the client, it expects to receive a response from the client within 30 seconds. If the client does not respond within the allotted time, the device retransmits the EAP-Request frame to the client. The time constraint for retransmission of EAP-Request frames to the client can be between 1 – 4294967295 seconds.

For example, to configure the device to retransmit an EAP-Request frame if the client does not respond within 45 seconds, enter the following command.

```
NetIron(config-dot1x)# supptimeout 45
```

Syntax: supptimeout <*seconds*>

Initializing 802.1x on a port

To initialize 802.1x port security on a port, or to flush all of its information on that port and start again, enter a command such as the following.

```
NetIron# dot1x initialize e 3/1
```

Syntax: dot1x initialize <*portnum*>

Allowing multiple 802.1x clients to authenticate

If there are multiple clients connected to a single 802.1x-enabled port, the device authenticates each of them individually. When multiple clients are connected to the same 802.1x-enabled port, the functionality described in "How 802.1x multiple client authentication works" on page 2066 is enabled by default. You can optionally do the following:

- Specify the authentication-failure action
- Specify the number of authentication attempts the device makes before dropping packets
- Disabling aging for dot1x-mac-sessions
- Configure aging time for blocked clients
- Clear the dot1x-mac-session for a MAC address

Specifying the authentication-failure action

In an 802.1x multiple client configuration, if RADIUS authentication for a client is unsuccessful, traffic from that client is either dropped in hardware (the default), or the client's port is placed in a "restricted" VLAN. You can specify which of these two authentication-failure actions is to be used. If the authentication-failure action is to place the port in a restricted VLAN, you can specify the ID of the restricted VLAN.

To specify that the authentication-failure action is to place the client's port in a restricted VLAN, enter the following command.

```
NetIron(config)# dot1x-enable
NetIron(config-dot1x)# auth-fail-action restricted-vlan
```

Syntax: [**no**] **auth-fail-action restricted-vlan**

To specify the ID of the restricted VLAN as VLAN 300, enter the following command.

```
NetIron(config-dot1x)# auth-fail-vlanid 300
```

Syntax: [**no**] **auth-fail-vlanid** <*vlan-id*>

Specifying the number of authentication attempts the device makes before dropping packets

When the initial authentication attempt made by the device to authenticate the client is unsuccessful, the device immediately retries to authenticate the client. After three unsuccessful authentication attempts, the client's802.1x MAC authentication session is set to either "access-denied" or the port is moved to restricted VLAN.

You can optionally configure the number of authentication attempts the device makes. To do so, enter a command such as the following.

```
NetIron(config-dot1x)# auth-fail-max-attempts 2
```

Syntax: [**no**] **auth-fail-max-attempts** <*attempts*>

By default, the device makes 3 attempts to authenticate a client. You can specify between 1 – 10 authentication attempts.

Display commands

The **show port security global-deny** command lists all the configured global deny MAC addresses.

The **show port security denied-macs** command lists all the denied MAC addresses in the system.

Clearing a dot1x-mac-session for a MAC address

You can clear the dot1x-mac-session for a specified MAC address, so that the client with that MAC address can be re-authenticated by the RADIUS server.

```
NetIron# clear dot1x mac-session 00e0.1234.abd4
```

Syntax: **clear dot1x mac-session** <*mac-address*>

Displaying 802.1x information

You can display the following 802.1x-related information:

- Information about the 802.1x configuration on the device and on individual ports
- Statistics about the EAPOL frames passing through the device
- Information about 802.1x-enabled ports dynamically assigned to a VLAN
- Information about the user-defined and dynamically applied Mac address and IP ACLs currently active on the device
- Information about the 802.1x multiple client configuration

Displaying 802.1x configuration information

To display information about the 802.1x configuration on the device, enter the following command.

```
NetIron# show dot1x
PAE Capability                : Authenticator Only
system-auth-control           : Enable
Number of ports enabled       : 25
re-authentication             : Disable
global-filter-strict-security: Enable
quiet-period                  : 60 Seconds
tx-period                     : 30 Seconds
supptimeout                   : 30 Seconds
servertimeout                 : 30 Seconds
maxreq                        : 3
re-authperiod                 : 3600 Seconds
Protocol Version              : 1
auth-fail-action              : Block Traffic
MAC Session Aging             : All
MAC Session Max Age           : 120 Seconds
Maximum Failed Attempts       : 3
```

Syntax: show dot1x

The following table describes the information displayed by the **show dot1x** command.

TABLE 398 Output from the **show dot1x command**

This field...	Displays...
PAE Capability	The Port Access Entity (PAE) role for the device. This is always “Authenticator Only”.
system-auth-control	Whether system authentication control is enabled on the device. The **dot1x-enable** command enables system authentication control on the device.
Number of ports enabled	Number of interfaces on the devices that have been enabled for 802.1x.
re-authentication	Whether periodic re-authentication is enabled on the device. Refer to “Configuring periodic re-authentication” on page 2076. When periodic re-authentication is enabled, the device automatically re-authenticates clients every 3,600 seconds by default.
global-filter-strict-security	Whether or not strict security mode is enabled globally.
quiet-period	When the device is unable to authenticate a client, the amount of time the device waits before trying again (default 60 seconds). Refer to “Setting the quiet period” on page 2077 for information on how to change this setting.
tx-period	When a client does not send back an EAP-response or identity frame, the amount of time the device waits before retransmitting the EAP-request or identity frame to a client (default 30 seconds). Refer to “Setting the interval for retransmission of EAP-request or identity frames” on page 2077 for information on how to change this setting.
supp-timeout	When a client does not respond to an EAP-request frame, the amount of time before the device retransmits the frame. Refer to “Specifying a timeout for retransmission of EAP-request frames to the client” on page 2078 for information on how to change this setting.
server-timeout	When the Authentication Server does not respond to a message sent from the client, the amount of time before the device retransmits the message. Refer to “Specifying a timeout for retransmission of messages to the Authentication Server” on page 2078 for information on how to change this setting.

TABLE 398 Output from the **show dot1x command** (Continued)

This field...	Displays...
max-req	The number of times the device retransmits an EAP-request or identity frame if it does not receive an EAP-response or identity frame from a client (default 2 times). Refer to "Specifying the number of EAP-request or identity frame retransmissions" on page 2078 for information on how to change this setting.
re-authperiod	How often the device automatically re-authenticates clients when periodic re-authentication is enabled (default 3,600 seconds). Refer to "Configuring periodic re-authentication" on page 2076 for information on how to change this setting.
security-hold-time	This field is not supported.
Protocol Version	The version of the 802.1x protocol in use on the device.
Auth-fail-action	The configured authentication-failure action. This can be Restricted VLAN or Block Traffic.
Mac Session Aging	Whether aging for dot1x-mac-sessions has been enabled or disabled for permitted or denied dot1x-mac-sessions.
Mac Session max-age	The configured software aging time for dot1x-mac-sessions.
Maximum Failed Attempts	The number of failed authentication attempts, if the authentication-failure action shows Restricted VLAN,

To display information about the 802.1x configuration on an individual port, enter a command such as the following.

```
NetIron# show dot1x config e 1/3

Port 1/3 Configuration:
AuthControlledPortControl    : Auto
max-clients                  : 32
multiple-clients             : Enable
filter-strict-security       : Enable
```

Syntax: show dot1x config ethernet <*slot/port*>

The following additional information is displayed in the **show dot1x config** command for an interface.

TABLE 399 Output from the **show dot1x config command** for an interface

This field...	Displays...
AuthControlledPortControl	The port control type configured for the interface. If set to auto, authentication is activated on the 802.1x-enabled interface.
multiple-hosts	Whether the port is configured to allow multiple Supplicants accessing the interface on the device through a hub. Refer to "Allowing multiple 802.1x clients to authenticate" on page 2079 for information on how to change this setting.
max-clients	The maximum number of clients that can be authenticated on this interface.
multiple-clients	Shows if the interface is enabled or disabled for multiple client authentication.
filter-strict-security	Shows if the interface is enabled or disabled for strict security mode.

Displaying 802.1x statistics

To display 802.1x statistics for an individual port, enter a command such as the following.

```
NetIron# show dot1x statistics e 3/3

Port 1/3 Statistics:
RX EAPOL Start:                 0
RX EAPOL Logoff:                0
RX EAPOL Invalid:               0
RX EAPOL Total:                 2
RX EAP Resp/Id:                 1
RX EAP Resp other than Resp/Id: 1
RX EAP Length Error:            0
Last EAPOL Version:             1
Last EAPOL Source:              0050.da0b.8bef
TX EAPOL Total:                 3
TX EAP Req/Id:                  1
TX EAP Req other than Req/Id:   1
Num Sessions:                   1
Num Restricted Sessions:        0
Num Authorized Sessions:        1
```

Syntax: show dot1x statistics [**all** | **ethernet** <*slot/port*>]

The following table describes the information displayed by the **show dot1x statistics** command for an interface.

TABLE 400 Output from the **show dot1x statistics command**

This field...	Displays...
RX EAPOL Start	The number of EAPOL-Start frames received on the port.
RX EAPOL Logoff	The number of EAPOL-Logoff frames received on the port.
RX EAPOL Invalid	The number of invalid EAPOL frames received on the port.
RX EAPOL Total	The total number of EAPOL frames received on the port.
RX EAP Resp or Id	The number of EAP-Response or Identity frames received on the port
RX EAP Resp other than Resp or Id	The total number of EAPOL-Response frames received on the port that were not EAP-Response or Identity frames.
RX EAP Length Error	The number of EAPOL frames received on the port that have an invalid packet body length.
Last EAPOL Version	The version number of the last EAPOL frame received on the port.
Last EAPOL Source	The source MAC address in the last EAPOL frame received on the port.
TX EAPOL Total	The total number of EAPOL frames transmitted on the port.
TX EAP Req or Id	The number of EAP-Request or Identity frames transmitted on the port.
TX EAP Req other than Req or Id	The number of EAP-Request frames transmitted on the port that were not EAP-Request or Identity frames.
Num sessions	Total number of dot1x sessions, which include authenticated, restricted, denied and sessions in the initial state.

TABLE 400 Output from the **show dot1x statistics command** (Continued)

This field...	Displays...
Num Restricted Sessions	Number of current 802.1x sessions that failed authentication. The user configuration was moved into a restricted VLAN.
Num Authorized Sessions	Number of current 802.1x authenticated sessions that are authorized.

Clearing 802.1x statistics

You can clear the 802.1x statistics counters on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to clear the 802.1x statistics counters on all interfaces on the device, enter the following command.

```
NetIron# clear dot1x statistics all
```

Syntax: clear dot1x statistics all

To clear the 802.1x statistics counters on interface e 3/11, enter the following command.

```
NetIron# clear dot1x statistics e 3/11
```

Syntax: clear dot1x statistics [**mac-address** | **ethernet** *<slot>/<portnum>*]

Displaying dynamically assigned VLAN information

The **show interface** command displays the VLAN to which an 802.1x-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port's default VLAN).

The following is an example of the **show interface** command indicating the port's dynamically assigned VLAN. Information about the dynamically assigned VLAN is shown in bold type.

```
NetIron# show interface e 12/2
GigabitEthernet1/3 is up, line protocol is up
  Hardware is GigabitEthernet, address is 000c.dbe2.5800 (bia 000c.dbe2.5800)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of L2 VLAN ID 4094 (dot1x-RADIUS assigned), original L2 VLAN ID is 1,
  port is untagged, port state is Forwarding
  STP configured to ON, Priority is level0, flow control enabled
  Force-DSCP disabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Internet address is 12.12.12.250/24, MTU 1522 bytes, encapsulation ethernet
  300 second input rate: 810 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 1253 bits/sec, 1 packets/sec, 0.00% utilization
  70178 packets input, 7148796 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 70178 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 70178 packets
  91892 packets output, 10081165 bytes, 0 underruns
  Transmitted 9853 broadcasts, 13330 multicasts, 68709 unicasts
  0 output errors, 0 collisions, DMA transmitted 91892 packets
```

In this example, the 802.1x-enabled port has been moved from VLAN 1 to VLAN 4094. When the client disconnects, the port will be moved back to VLAN 1.

Displaying information on MAC address filters and IP ACLs on an interface

You can display information about the user-defined and dynamically applied MAC address filters and IP ACLs currently active on an interface.

Displaying MAC address filters applied to an 802.1x-enabled port

Use the **show dot1x mac-address** command to display information about MAC filters applied to an interface. If the MAC address filter is dynamically assigned by 802.1x, the display shows the following.

```
NetIron#show dot1x mac-address ethernet 1/1
Port 1/1 MAC Address Filter information:
  802.1x dynamic MAC Filter (user defined) :
    mac access-list 401 in
  Port default MAC Filter :
    mac access-list 400 in
```

The “Port default MAC Filter” appears if a default MAC filter has been configured on the port. This default MAC filter is the MAC filter that will be applied to the port once the dynamically assigned MAC filter is removed. If a default MAC filter has not been configured, the message “No Port default MAC is displayed.

When the dynamically assigned MAC address filter is removed, the display shows the following information.

```
NetIron#show dot1x mac-address ethernet 1/1
```

```
Port 1/1 MAC Address Filter information:
  Port default MAC Filter :
    mac access-list 400 in
```

Syntax: **show dot1x mac-address-filter** [**all** | **ethernet** <*slot/port*> | | **begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

The **all** keyword displays all dynamically applied MAC address filters active on the device.

Use the **ethernet** <*slot*>/<*port*> parameter to display information for one port.

Displaying IP ACLs applied to an 802.1x-enabled port

Use the **show dot1x ip-acl** command to display the information about what IP ACLs have been applied to an 802.1x-enabled port. If the IP ACL was dynamically applied by 802.1x, the following information is displayed.

```
NetIron# show dot1x ip-acl ethernet 1/1
Port 1/1 IP ACL information:
  802.1x dynamic IP ACL (user defined) in:
    ip access-list extended Port_1/1_E_IN in
  Port default IP ACL in:
    ip access-list 100 in
  No outbound ip access-list is set
```

The “Port default IP ACL” appears if a default IP ACL has been configured on the port. The default IP ACL is the IP ACL that will be applied to the port once the dynamically assigned IP ACL is removed. If a default IP ACL has not been configured, the message “No Port default IP ACL” is displayed.

When the dynamically assigned IP ACL is removed from the port, the display shows the following information.

```
NetIron# show dot1x ip-acl ethernet 1/1
Port 1/1 IP ACL information:
  Port default IP ACL in:
    ip access-list 100 in
  No outbound ip access-list is set
```

Syntax: **show dot1x ip-acl [all | ethernet** <*slot/port*> | | **begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

The **all** keyword displays all dynamically applied IP ACLs active on the device.

Use the **ethernet** <*slot*>/<*port*> parameter to display information for one port.

Displaying information about the dot1x-mac-sessions on each port

To display information about the dot1x-mac-sessions on each port on the device, enter the following command.

```
NetIron# show dot1x mac-session
Port  MAC             Username                      VLAN Auth State ACL|MAC  Age
                                                                    i|o|f
-------------------------------------------------------------------------------
1/1   0050.da0b.8cd7  Mary M                          1    DENIED  n|n|n   0
1/2   0050.da0b.8cb3  adminmorn                    4094 PERMITTED  y|n|n   0
1/3   0050.da0b.8bef  reports                      4094 PERMITTED  y|n|n   0
1/4   0010.5a1f.6a63  testgroup                    4094 PERMITTED  y|n|n   0
1/5   0050.da1a.ff7e  admineve                     4094 PERMITTED  y|n|n   0
```

Syntax: **show dot1x mac-session** [**brief** | [**begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]]

Table 401 describes the information displayed by the **show dot1x mac-session** command.

TABLE 401 Output from the **show dot1x mac-session** command

This field...	Displays...
Port	The port on which the dot1x-mac-session exists.
MAC	The MAC address of the client
Username	The username used for RADIUS authentication.
Vlan	The VLAN to which the port is currently assigned.
Auth-State	The authentication state of the dot1x-mac-session. This can be one of the following: permit – The client has been successfully authenticated, and traffic from the client is being forwarded normally. blocked – Authentication failed for the client, and traffic from the client is being dropped in hardware. restricted – Authentication failed for the client, but traffic from the client is allowed in the restricted VLAN only. init - The client is in is in the process of 802.1x authentication, or has not started the authentication process.
ACL	Whether or not an IP ACL is applied to incoming (i) and outgoing (o) traffic on the interface
MAC	Whether or not a MAC filter is applied to the port.
Age	The software age of the dot1x-mac-session.

Displaying information about the ports in an 802.1x multiple client configuration

To display information about the ports in an 802.1x multiple client configuration, enter the following command.

```
NetIron# show dot1x mac-session brief
Port             Number of users      Dynamic Dynamic     Dynamic
          Restricted Authorized Total VLAN    ACL (In/Out)MAC-Filt
---------+----------+----------+-----+-------+-----------+--------
1/1               0          0     1 no          no/no   no
1/2               0          1     1 yes        yes/no   no
1/3               0          1     1 yes        yes/no   no
1/4               0          1     1 yes        yes/no   no
1/5               0          1     1 yes        yes/no   no
```

Syntax: **show dot1x mac-session brief** [| **begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

The following table describes the information displayed by the **show dot1x mac-session brief** command.

TABLE 402 Output from the **show dot1x mac-session brief** command

This field...	Displays...
Port	Information about the users connected to each port.
Number of users	The number of restricted and authorized (those that were successfully authenticated) users connected to the port.
Dynamic VLAN	Whether or not the port is a member of a RADIUS-specified VLAN.
Dynamic ACL	Whether or not a RADIUS-specified ACL has been applied to the port for incoming (in) and outgoing (out) traffic.
Dynamic MAC Filters	Whether or not a RADIUS-specified MAC Filter has been applied to the port.

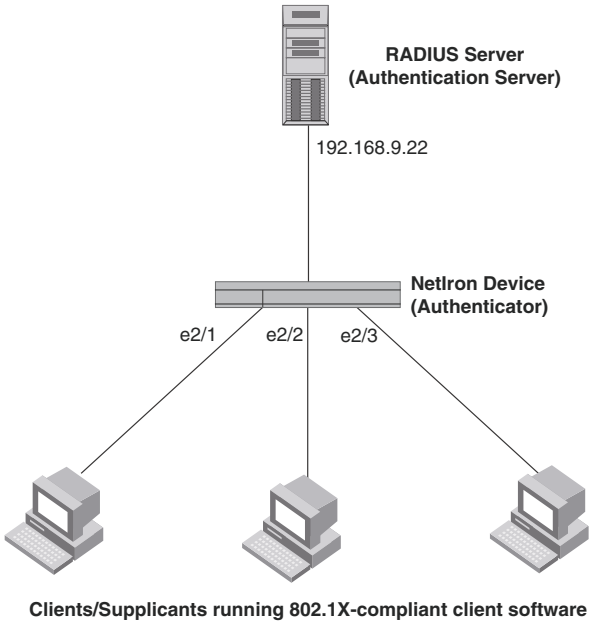
Sample 802.1x configurations

This section illustrates a sample point-to-point configuration and a sample hub configuration that use 802.1x port security.

Point-to-point configuration

Figure 229 illustrates a sample 802.1x configuration with clients connected to three ports on the device. In a point-to-point configuration, only one 802.1x client can be connected to each port.

FIGURE 229 Sample point-to-point 802.1x configuration



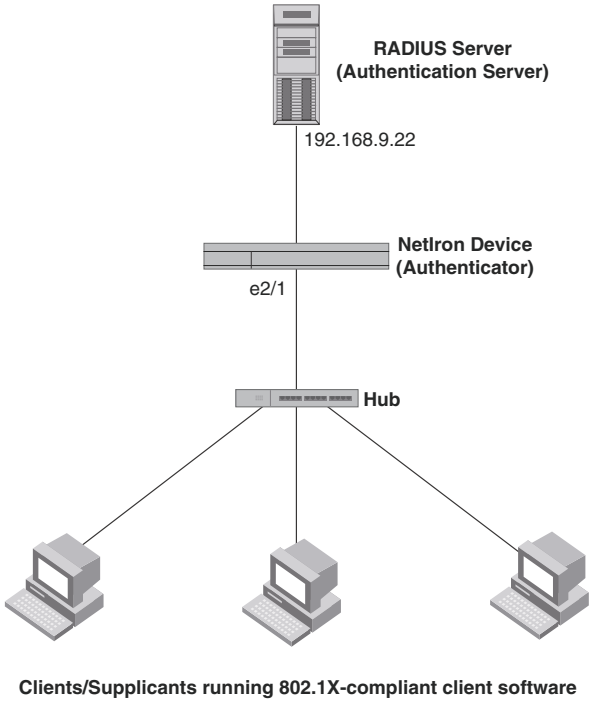
The following commands configure the device in Figure 229.

```
NetIron(config)# aaa authentication dot1x default radius
NetIron(config)# radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x
NetIron(config)# dot1x-enable e 2/1 to 2/3
NetIron(config-dot1x)# re-authentication
NetIron(config-dot1x)# timeout re-authperiod 2000
NetIron(config-dot1x)# timeout quiet-period 30
NetIron(config-dot1x)# timeout tx-period 60
NetIron(config-dot1x)# max-req 6
NetIron(config-dot1x)# exit
NetIron(config)# interface e 2/1
NetIron(config-if-e100-1)# dot1x port-control auto
NetIron(config-if-e100-1)# exit
NetIron(config)# interface e 2/2
NetIron(config-if-e100-2)# dot1x port-control auto
NetIron(config-if-e100-2)# exit
NetIron(config)# interface e 2/3
NetIron(config-if-e100-3)# dot1x port-control auto
NetIron(config-if-e100-3)# exit
```

Hub configuration

Figure 230 illustrates a configuration where three 802.1x-enabled clients are connected to a hub, which is connected to a port on the device. The configuration is similar to that in Figure 229, except that 802.1x port security is enabled on only one port, and the **multiple-hosts** command is used to allow multiple clients on the port.

FIGURE 230 Sample 802.1x configuration using a hub



The following commands configure the device in Figure 230.

```
NetIron(config)# aaa authentication dot1x default radius
NetIron(config)# radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x
NetIron(config)# dot1x-enable e 2/1
NetIron(config-dot1x)# re-authentication
NetIron(config-dot1x)# timeout re-authperiod 2000
NetIron(config-dot1x)# timeout quiet-period 30
NetIron(config-dot1x)# timeout tx-period 60
NetIron(config-dot1x)# max-req 6
NetIron(config-dot1x)# exit
NetIron(config)# interface e 2/1
NetIron(config-if-e100-1)# dot1x port-control auto
NetIron(config-if-e100-1)# exit
```

Chapter

Protecting against Denial of Service Attacks 52

PowerConnect B-MLXe supports the following Denial of Service (DoS) attack features they support.

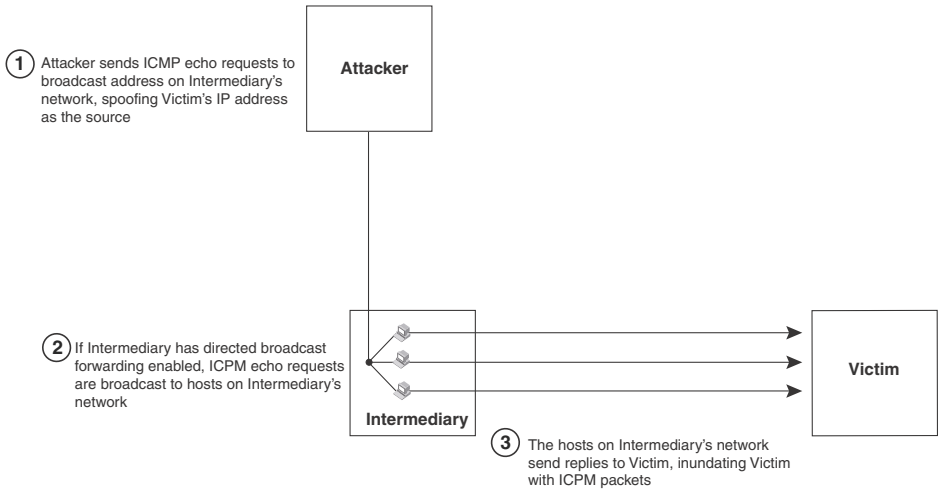
- Denial of Service (DoS)
- Protection Against smurf Attacks
- Protection Against TCP SYN Attacks
- Protection Against TCP Reset Attacks

In a DoS attack, a router is flooded with useless packets for the purpose of slowing down or stopping normal operation. Dell devices include measures to defend against two types of DoS attacks: Smurf attacks and TCP SYN attacks.

Protecting against smurf attacks

A ***smurf attack*** is a kind of DoS attack where an attacker causes a victim to be flooded with ICMP echo (pPing) replies sent from another network. Figure 231 illustrates how a smurf attack works.

FIGURE 231 How a smurf attack floods a victim with ICMP replies



The attacker sends an ICMP echo request packet to the broadcast address of an intermediary network. The ICMP echo request packet contains the spoofed address of a victim network as its source. When the ICMP echo request reaches the intermediary network, it is converted to a Layer 2 broadcast and sent to the hosts on the intermediary network. The hosts on the intermediary network then send ICMP replies to the victim network.

For each ICMP echo request packet sent by the attacker, a number of ICMP replies equal to the number of hosts on the intermediary network are sent to the victim. If the attacker generates a large volume of ICMP echo request packets, and the intermediary network contains a large number of hosts, the victim can be overwhelmed with ICMP replies.

Avoiding being an intermediary in a smurf attack

A smurf attack relies on the intermediary to broadcast ICMP echo request packets to hosts on a target subnet. When the ICMP echo request packet arrives at the target subnet, it is converted to a Layer 2 broadcast and sent to the connected hosts. This conversion takes place only when directed broadcast forwarding is enabled on the device.

To avoid being an intermediary in a smurf attack, make sure forwarding of directed broadcasts is disabled on the device. Directed broadcast forwarding is disabled by default. To disable directed broadcast forwarding, enter this command.

```
NetIron(config)# no ip directed-broadcast
```

Syntax: [**no**] **ip directed-broadcast**

Avoiding being a victim in a smurf attack

You can configure the device to drop ICMP packets when excessive numbers are encountered, as is the case when the device is the victim of a smurf attack. The following example sets threshold values for ICMP packets targeted at the router and drop them when the thresholds are exceeded.

```
NetIron(config)# ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

Syntax: ip icmp burst-normal <*value*> **burst-max** <*value*> **lockup** <*seconds*>

The **burst-normal** value can be from 1 – 100000.

The **burst-max** value can be from 1 – 100000.

The **lockup** value can be from 1 – 10000.

The number of incoming ICMP packets per second are measured and compared to the threshold values, as follows:

- If the number of ICMP packets exceeds the **burst-normal** value, the excess ICMP packets are dropped.
- If the number of ICMP packets exceeds the **burst-max** value, all ICMP packets are dropped for the number of seconds specified by the lockup value. When the lockup period expires, the packet counter is reset and measurement is restarted.

  In this example, if the number of ICMP packets received per second exceeds 5,000, the excess packets are dropped. If the number of ICMP packets received per second exceeds 10,000, the device drops all ICMP packets for the next 300 seconds (five minutes).

When incoming ICMP packets exceed the **burst-max** value, the following message is logged.

```
<date> <time>:N:Local ICMP exceeds <burst-max> burst packets, stopping for
<lockup> seconds!!
```

IPv6 traffic not subject to DOS attack filtering

The following IPv6 traffic exceptions (per section 4.4 of RFC 4890) are not subject to DoS attack filtering.

Error messages that are essential to the establishment and maintenance of communications:

- Destination unreachable (Type 1) - All codes
- Packet Too Big (Type 2)
- Time Exceeded (Type 3) - Code 0 only
- Parameter Problem (Type 4) - Codes 1 and 2 only

Address configuration and router selection messages:

- Router Solicitation (Type 133)
- Router Advertisement (Type 134)
- Neighbor Solicitation (Type 135)
- Neighbor Advertisement (Type 136)
- Inverse Neighbor Discovery Solicitation (Type 141)
- Inverse Neighbor Discovery Advertisement (Type 142)

Link-local multicast receiver notification messages:

- Listener Query (Type 130)
- Listener Report (Type 131)
- Listener Done (Type 132)
- Listener Report v2 (Type 143)

Multicast Router Discovery messages:

- Multicast router advertisement (Type 151)
- Multicast router solicitation (Type 152)
- Multicast router termination (Type 153)

Section 4.4 of RFC 4890 also recommends that the following traffic types must not be dropped, however these traffic types will continue to be subject to DoS attack filtering:

- Echo request (Type 128)
- Echo response (Type 129)
- Certificate path solicitation (Type 148)
- Certificate path advertisement (Type 149)

Protecting against TCP SYN attacks

TCP SYN attacks disrupt normal traffic flow by exploiting the way TCP connections are established. When a TCP connection starts, the connecting host sends a TCP SYN packet to the destination host. The destination host responds with a SYN ACK packet, and the connecting host sends back an ACK packet. This process, known as a “TCP three-way handshake”, establishes the TCP connection.

While waiting for the connecting host to send an ACK packet, the destination host keeps track of the as-yet incomplete TCP connection in a connection queue. When the ACK packet is received, information about the connection is removed from the connection queue. Usually there is not much time between the destination host sending a SYN ACK packet and the source host sending an ACK packet, so the connection queue clears quickly.

In a TCP SYN attack, an attacker floods a host with TCP SYN packets that have random source IP addresses. For each of these TCP SYN packets, the destination host responds with a SYN ACK packet and adds information to the connection queue. However, since the source host does not exist, no ACK packet is sent back to the destination host, and an entry remains in the connection queue until it ages out (after approximately one minute). If the attacker sends enough TCP SYN packets, the connection queue can fill up, and service can be denied to legitimate TCP connections.

To protect against TCP SYN attacks, you can configure Dell devices to drop TCP SYN packets when excessive numbers are encountered. You can set threshold values for TCP SYN packets that are targeted at the device and drop them when the thresholds are exceeded, as shown in this example.

```
NetIron(config)# ip tcp burst-normal 10 burst-max 100 lockup 300
```

Syntax: **ip tcp burst-normal** <*value*> **burst-max** <*value*> **lockup** <*seconds*>

The **burst-normal** value can be from 1 – 100000.

The **burst-max** value can be from 1 – 100000.

The **lockup** value can be from 1 – 10000.

The number of incoming TCP SYN packets per second is measured and compared to the threshold values as follows:

- If the number of TCP SYN packets exceeds the **burst-normal** value, the excess TCP SYN packets are dropped.
- If the number of TCP SYN packets exceeds the **burst-max** value, all TCP SYN packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

  In this example, if the number of TCP SYN packets received per second exceeds 10, the excess packets are dropped. If the number of TCP SYN packets received per second exceeds 100, the device drops all TCP SYN packets for the next 300 seconds (five minutes).

When incoming TCP SYN packets exceed the burst-max value, the following message is logged.

```
<date> <time>:N:Local TCP exceeds <burst-max> burst packets, stopping for <lockup>
seconds!!
```

TCP security enhancement

A TCP security enhancement improves the way TCP inbound segments are handled. This enhancement eliminates or minimizes the possibility of a TCP reset attack, in which a perpetrator attempts to prematurely terminate an active TCP session, and a data injection attack, where an attacker injects or manipulates data in a TCP connection.

In both cases, the attack is blind, meaning the perpetrator does not have visibility into the content of the data stream between two devices, but blindly injects traffic. The attacker also does not see the direct effect (the continuing communications between the devices and the impact of the injected packet) but may see the indirect impact of a terminated or corrupted session.

The TCP security enhancement prevents and protects against the following types of attacks:

- Blind TCP reset attack using the reset (RST) bit.
- Blind TCP reset attack using the synchronization (SYN) bit
- Blind TCP packet injection attack

The TCP security enhancement is automatically enabled. If necessary, you can disable this feature. Refer to "Disabling the TCP security enhancement" on page 2095.

Protecting against a blind TCP reset attack using the RST bit

In a blind TCP reset attack using the RST bit, a perpetrator attempts to guess the RST segments to prematurely terminate an active TCP session.

To prevent a user from using the RST bit to reset a TCP connection, the RST bit is subject to the following rules when receiving TCP segments:

- If the RST bit is set and the sequence number is outside the expected window, the device silently drops the segment.
- If the RST bit is exactly the next expected sequence number, the device resets the connection.
- If the RST bit is set and the sequence number does not exactly match the next expected sequence value, but is within the acceptable window, the device sends an acknowledgement (ACK).

The TCP security enhancement is enabled by default. To disable it, refer to "Disabling the TCP security enhancement" on page 2095.

Protecting against a blind TCP reset attack using the SYN bit

For a blind TCP reset attack, the attacker tries to guess the SYN bits to terminate an active TCP session.To protect against this type of attack, the SYN bit is subject to the following rules during arrival of TCP segments:

- If the SYN bit is set and the sequence number is outside the expected window, the device sends an ACK to the peer.
- If the SYN bit is set and the sequence number is an exact match to the next expected sequence, the device sends an ACK segment to the peer. Before sending the ACK segment, the software subtracts a 1 from the value being acknowledged.
- If the SYN bit is set and the sequence number is acceptable, the device sends an ACK segment to the peer.

This TCP security enhancement is enabled by default. To disable it, refer to "Disabling the TCP security enhancement" on page 2095.

Protecting against a blind injection attack

In a blind TCP injection attack, the attacker tries to inject or manipulate data in a TCP connection. To reduce the chances of a blind injection attack, an additional check is performed on all incoming TCP segments.

This TCP security enhancement is enabled by default. To disable it, refer to "Disabling the TCP security enhancement" on page 2095.

Disabling the TCP security enhancement

The TCP security enhancement is enabled by default. If necessary, you can disable this feature. When you disable this feature, the device reverts to the original behavior.

To disable the TCP security enhancement, enter the following command at the Global CONFIG level of the CLI.

```
NetIron(config)# no ip tcp tcp-security
```

To re-enable the TCP security enhancement after it has been disabled, enter the following command.

```
NetIron(config)# ip tcp tcp-security
```

Syntax: [**no**] **ip tcp tcp-security**

Displaying statistics from a DoS attack

You can display statistics about ICMP and TCP SYN packets that were dropped, passed, or blocked because burst thresholds were exceeded using the **show statistics dos-attack** command.

```
NetIron# show statistics dos-attack
------ Local Attack Statistics ------
ICMP Drop Count     Port Block Count    SYN Drop Count    SYN Block Count
---------------     ----------------    --------------    ---------------
              0                   0                 0                  0
```

Syntax: **show statistics dos-attack** [| **begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

Table 403 describes this output.

TABLE 403 Output from the **show statistics dos-attack** command

This field...	Displays...
Packet drop count	Number of packets that are dropped when the port is in lockup mode.
Packet Pass Count	Number of packets that are forwarded when the port is in rate-limiting mode.
Packet BLock Count	Number of times the port was shut down for a traffic flow that matched the ACL.

Clear DoS attack statistics

To clear statistics about ICMP and TCP SYN packets, enter the **clear statistics dos-attack** command.

```
NetIron(config)# clear statistics dos-attack
```

Syntax: **clear statistics dos-attack**

Chapter

Reverse Path Forwarding 53

The following Reverse Path Forwarding features are supported by PowerConnect B-MLXe Series.

- Reverse Path Forwarding (RPF)
- RPF Support for IP over MPLS Routes
- Suppressing RPF for Packets Using inbound ACLs
- Excluding Packets that Match the Routers Default Route

A number of common types of denial-of-service (DoS) attacks, including Smurf and Tribe Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. Reverse Path Forwarding (RPF) is designed to prevent such a malicious user from spoofing a source IP address by checking that the source address specified for a packet is received from a network that the PowerConnect has access to. Packets with invalid source addresses are not forwarded. Optionally, you can log packets that fail the RPF test.

RPF is supported for IPv6 packets. Differences in RPF support in IPv4 and IPv6 are noted within this chapter where necessary.

Configuration of RPF

To configure the PowerConnect for RPF, you must perform the following steps:

- Considerations for Configuring RPF
- Special Considerations for Configuring RPF with ECMP routes
- RPF Support for IP over MPLS routes
- RPF Compatible CAM profiles
- Configure the global command: **reverse-path-check**
- Enable RPF mode individually on ports that you want it to run.
- Suppressing RPF for Packets with Specified Address Prefixes

Configuration considerations for RPF

You must consider the following when configuring RPF:

- IP packets with source IP address of 0.0.0.0 will always fail RPF check
- If you attempt to enable the global RPF command on a system with incompatible CAM settings, the command will be rejected and you will receive a console message describing this.
- Since the RPF feature requires that the entire IP route table is available in hardware, the feature must work in conjunction with Foundry Direct Routing (FDR). FDR is the default mode of operation for the PowerConnect. For more information about enabling and disabling FDR refer to “Foundry Direct Routing” on page 40-1.

- You cannot configure RPF on a physical port that has VRF configured on it or if the physical port belongs to a virtual interface with a VRF configuration.
- Only RPF loose mode is supported for GRE routes.
- If a default route is present on the router, loose mode will permit all traffic.
- RPF can only be configured at the physical port level. It should not be configured on virtual interfaces.
- IPv6 packets with a link-local source address are not subject to IPv6 RPF check.
- IPv6 RPF check is not supported for 6-to-4 tunnel routes.

Special considerations for configuring RPF with ECMP routes

RPF for IPv6 is not subject to the special considerations for configuring RPF with ECMP routes described here.

For a source IP address matching an ECMP route, RPF will permit the packet if it arrives on any of the next-hop interfaces for that route. For example, if there are two best next-hops for a network route 11.11.11.0/24, one pointing to 10.10.10.1 (Gigabit Ethernet 7/1) and the other to 10.10.30.1 (Gigabit Ethernet 7/12), then incoming packets with source address matching 11.11.11.0/24 will be permitted on either Gigabit Ethernet 7/1 or Gigabit Ethernet 7/12.

A disadvantage of this configuration is that if some other route shares any of these next-hops, the packets with a source IP address matching that route will also be permitted from any of the interfaces associated with those next hops. For example, say 12.12.12.0/24 has the next-hop 10.10.10.1, then packets from 12.12.12.0/24 will also be be permitted on either Gigabit Ethernet 7/1 or Gigabit Ethernet 7/12.

RPF support for IP over MPLS routes

For IPv4 routes over MPLS tunnels, the physical interface for an outgoing tunnel on which a route is assigned may not be the same as the one from which we receive packets. Consequently, only RPF loose mode is supported on MPLS uplinks. IPv6 is not currently supported over MPLS. When it is supported, it will only support RPF loose mode on MPLS uplinks.

RPF compatible CAM profiles

Not all CAM profiles are compatible with RPF. Table 404 lists all of the RPF Compatible CAM profiles by software release. Refer to "CAM partition profiles" on page 2240 for a description of each of the available CAM profiles.

TABLE 404 RPF compatible and non-compatible CAM profiles

Software release	Compatible CAM profiles	Non-compatible CAM profiles
MLX 03.2.00 and later	default	ipv4-ipv6
	ipv4	ipv4-vpls
	ipv4-vpn	l2-metro
	ipv6	l2-metro-2

TABLE 404 RPF compatible and non-compatible CAM profiles

Software release	Compatible CAM profiles	Non-compatible CAM profiles
	mpls-l3vpn	mpls-vpls
	mpls-l3vpn-2	mpls-vpls-2
	ipv4-ipv6-2	mpls-vpn-vpls
	multi-service-2 (added in release 03.8.00)	multi-service
MLX 03.0.00 and 03.1.00	ipv4	default
	ipv6	ipv4-vpn
		mpls-l3vpn
		mpls-l3vpn-2
		ipv4-ipv6
		ipv4-vpls
		l2-metro
		l2-metro-2
		mpls-vpls
		mpls-vpls-2
		mpls-vpn-vpls
		multi-service
All IMR Releases	default	ipv4
	ipv4-ipv6	ipv4-vpls
	ipv6	ipv4-vpn
	l2-metro	l2-metro-2
	mpls-l3vpn	mpls-l3vpn-2
	mpls-vpls	mpls-vpls-2
	rpf	mpls-vpn-vpls
		multi-service

Configuring the global RPF command

Before you can enable RPF to operate on a PowerConnect you must first configure RPF globally. There are separate command for IPv4 and IPv6, as shown in the following.

For IPv4.

```
NetIron(config)# reverse-path-check
```

Syntax: reverse-path-check

```
NetIron(config)# ipv6 reverse-path-check
```

For IPv6.

Syntax: ipv6 reverse-path-check

Enable RPF on individual ports

After RPF has been configured globally for a PowerConnect, it must be configured on every interface that you want it to operate. The RPF feature can only be configured on physical ethernet interfaces. There are two modes "strict" and "loose" that can be configured to enforce RPF on IP addresses for packets arriving on a given interface, as described in the following:

- In **loose** mode, RPF will permit a packet as long as the source address matches a known route entry in the routing table. It will drop a packet if it does not match a route entry. Please note that if a default route is present, loose mode will permit all traffic.
- In **strict** mode, RPF requires that a packet matches a known route entry as described in loose mode and also that it arrives at the interface as described in the router table's next hop information. It will drop a packet that does not match both of these criteria.

Configuring RPF on a port requires a separate command for IPv4 and IPv6. To configure RPF on a port, use the IPv4 or IPv6 command, as shown in the following.

For IPv4.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e1000-3/1)# rpf-mode strict log
```

Syntax: [no] rpf-mode [loose | strict] [log]

For IPv6.

```
NetIron(config)# interface ethernet 3/1
NetIron(config-if-e1000-3/1)# rpf-mode-ipv6 strict log
```

Syntax: [no] rpf-mode-ipv6 [loose | strict] [log]

There are two modes in which you can enforce RPF on IP sources address for packets that arrive on a configured interface: **loose** mode and **strict** mode as described previously.

The **log** parameter directs RPF to log packets that fail the RPF test. Enabling RPF logging may lead to high CPU utilization on the interface module because packets that fail the RPF check test are dropped in software. Only syslog entries are created by this option. No SNMP traps are issued by this option.

The ACL or RPF logging mechanism on the Interface modules log a maximum of 256 messages per minute, and send these messages to the Management module. A rate-limiting mechanism has been added to rate-limit the number of messages from the Interface module CPU to the Management module CPU to 5 messages per second. Because this delays the delivery of messages to the Management module, in the worst case scenario with all 256 packets arriving at the same time on the Interface module, the time values stamped by the Management module on the messages will vary by as much as 60 seconds.

Configuring a timer interval for IPv6 session logging

You can use the **ipv6 session-logging-age** command to globally configure a timer interval for IPv6 session logging. The timer interval is set for 3 minutes in the following example.

```
NetIron(config)# ipv6 session-logging-age 3
```

Syntax: [no] ipv6 session-logging-age <*minutes*>

The <*minutes*> variable sets the timer interval for logging. Configurable variables are 1 - 10 minutes. The default value is 5 minutes.

You can use the **show syslog** command to view RPF messages as shown in the following.

```
NetIron# show syslog
Dec 18 19:32:52:I:IPv6 RPF: Denied 1 packet(s) on port 1/2 tcp fec0:1::2(0) ->
4500:1::2(0)
```

Suppressing RPF for packets with specified address prefixes

You can suppress RPF packet drops for a specified set of packets using inbound ACLs. To do this.

Create an IPv4 or IPv6 ACL that identifies the address range that you do not want dropped and specify the flag **suppress-rpf-drop** to the ACL clause. When a packet that fails the RPF check and matches the specified ACL permit clause with the suppress-rpf-drop flag set, it is forwarded as a normal packet and it is accounted as a **unicast RPF suppressed drop packet** as described in Table 405.

The following example demonstrates the configuration of the IPv4 ACL named **access-list 135** which permits traffic from the source network 4.4.4.0/24 even if the RPF check test fails.

```
NetIron(config)# access-list 135 permit ip 4.4.4.0.0.0.255 any
suppress-rpf-drop
NetIron(config)# access-list 135 permit ip any any
```

The following example demonstrates the configuration of the IPv6 ACL named **rpf1** which permits traffic from the source host 2002::1 even if the RPF check test fails.

```
NetIron(config)# ipv6 access-list rpf1
NetIron(config-ipv6-access-list rpf1)# permit tcp host 2002::1 any
suppress-rpf-drop
```

Syntax: suppress-rpf-drop

In the following example, the IPv4 ACL 135 is applied as an inbound filter on ethernet interface 7/5.

```
NetIron(config)# interface ethernet 7/5
NetIron(config-if-e1000-7/5)# rpf-mode strict
NetIron(config-if-e1000-3/1)# ip access-group 135 in
```

In the following example, the IPv6 ACL named “rpf1” is applied as an inbound filter on ethernet interface 7/5.

```
NetIron(config)# interface ethernet 7/5
NetIron(config-if-e1000-7/5)# ipv6-rpf-mode strict
NetIron(config-if-e1000-3/1)# ipv6 traffic-filter rpf1 in
```

NOTE
If the physical port is a member of a virtual interface, the ACL will have to be applied to the virtual interface instead of the physical port.

Excluding packets that match the routers default route

A new command was introduced that directs the PowerConnect router to drop packets whose source address match the routers default route and increment the RPF drop counter. Using this feature requires that RPF be configured globally first. This feature is configured separately for IPv4 and IPv6 as described in the following.

For IPv4.

```
NetIron(config)# reverse-path-check
NetIron(config)# urpf-exclude-default
```

Syntax: urpf-exclude-default

For IPv6.

```
NetIron(config)# ipv6 reverse-path-check
NetIron(config)# ipv6 urpf-exclude-default
```

Syntax: ipv6 urpf-exclude-default

Displaying RPF statistics

To display information about RPF configuration and packets that have been dropped because they failed the RPF check, use the **show ip interface** or the **show ipv6 interface** command as shown.

For IPv4.

```
NetIron# show ip interface ethernet 7/1
Interface Ethernet 7/1 (384)
    port enabled
    port state: UP
    ip address: 1.2.3.4/8
    Port belongs to VRF: default
    encapsulation: Ethernet, mtu: 1500
    MAC Address 000c.db24.a6c0
    directed-broadcast-forwarding: disabled
    No inbound ip access-list is set
    No outbound ip access-list is set
    No Helper Addresses are configured
    RPF mode: strict RFP Log: Disabled
    376720 unicast RPF drop 36068 unicast RPF suppressed drop
```

For IPv6:

```
NetIron#show ipv6 interface ethernet 3/1
Interface Ethernet  3/1 is down, line protocol is down
  IPv6 is enabled, link-local address is
  Global unicast address(es):
  Joined group address(es):
    ff02::2
    ff02::1
  MTU is 1500 bytes
  ICMP redirects are disabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30 seconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1 seconds
  ND advertised retransmit interval is 0 seconds
  ND router advertisements are sent every 200 seconds
```

```
ND router advertisements live for 1800 seconds
No Inbound Access List Set
No Outbound Access List Set
IPv6 RPF mode: Strict  IPv6 RPF Log: Enabled
RxPkts:       0                      TxPkts:   0
RxBytes:      0                      TxBytes:  0
IPv6 unicast RPF drop: 0
IPv6 unicast RPF suppressed drop: 0
```

NOTE
The RPF accounting information is always available through the physical interface, even if the physical port belongs to one or more VE's

Table 405 describes the RPF statistics displayed when using the **show ip interface** or **show ipv6 interface** command. They are displayed in **bold**.

TABLE 405 RPF statistics by port

This field...	Displays...
RPF Mode:	This display parameter can have one of the following two values: • loose – RPF will permit a packet as long as the source address matches a known route entry in the routing table. It will drop a packet if it does not match a route entry. • strict – RPF requires that a packet matches a known route entry as described for loose mode and also that it arrives at the configured interface as described in the router table's next hop information. It will drop a packet that does not match both of these criteria.
RPF Log:	This display parameter displays the RPF Log Configuration Status: • Enabled – the RPF log feature has been configured. • Disabled – the RPF log feature has not been configured
<*number*> unicast RPF drop	The number of packets that have been dropped due to failure of the RPF test.
<*number*> unicast RPF suppressed drop	The number of packets would have been dropped due to failure of the RPF test but were not dropped because they matched conditions set in an ACL with the **suppress-rpf-drop** flag set.

Clearing RPF statistics for a specified IPv4 interface

To clear RPF statistics on a specific IPv4 physical interface use the following command.

Syntax: **clear ip interface ethernet** <*slot/port*> **| pos** <*slot/port*>

Use the **ethernet** or **pos** parameters to specify whether the port is Ethernet or Packet-over-Sonet.

The slot/port variables specify the interface that you want to clear RPF statistics for.

Clearing RPF statistics for all IPv4 interfaces within a router

To clear RPF statistics on all IPv4 physical interfaces within a router, use the following command.

Syntax: **clear ip interface counters**

Clearing RPF statistics for a specified IPv6 interface

To clear RPF statistics on a specific IPv6 physical interface use the following command.

Syntax: clear ipv6 interface ethernet <*slot/port*> **| pos** <*slot/port*>

Use the **ethernet** or **pos** parameters to specify whether the port is Ethernet or Packet-over-Sonet.

The slot/port variables specify the interface that you want to clear RPF statistics for.

Clearing RPF statistics for all IPv6 interfaces within a router

To clear RPF statistics on all IPv6 physical interfaces within a router, use the following command.

Syntax: clear ipv6 interface counters

Displaying RPF logging

If you set the log option of the rpf-mode command, the packets are saved to the system log. To display the log, enter the following.

```
NetIron## show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 1305 overruns)
    Buffer logging: level ACDMEINW, 50 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
Dynamic Log Buffer (50 lines):
May 11 12:12:54:I:RPF: Denied 1 packets on port 7/5 tcp 4.4.4.1(0) -> 5.6.7.8(0)
```

NOTE
A maximum of 256 RPF log messages are logged per minute.

Chapter

Securing SNMP Access 54

Table displays the SNMPv3 features supported by PowerConnect B-MLXe Series.

- SNMP v3
- User-Based Security Mode
- Community Strings to Authenticate SNMP Access
- New encryption code for passwords, authentication keys, and community strings
- Defining an SNMP User Account
- AES for SNMPv3

Simple Network Management Protocol (SNMP) is a set of protocols for managing complex networks. SNMP sends messages, called protocol data units (PDUs), to different parts of a network. An SNMP-compliant device, called an agent, stores data about itself in Management Information Bases (MIBs) and SNMP requesters or managers.

NOTE
SNMP agent is disabled by default. Use the **snmp-server** command to enable the agent.

Establishing SNMP community strings

SNMP versions 1 and 2c use community strings to restrict SNMP access. The default passwords for SNMP access are the SNMP community strings configured on the device:

- The default read-only community string is “public”. Use this community string for any SNMP Get, GetNext, or GetBulk request.
- By default, you cannot perform any SNMP Set operations since a read-write community string is not configured.

You can configure as many additional read-only and read-write community strings as you need. The number of strings you can configure depends on the memory on the device. There is no practical limit.

If you delete the startup configuration file, the device automatically re-adds the default “public” read-only community string the next time you load the software.

Encryption of SNMP community strings

Encryption is enabled by default. The software automatically encrypts SNMP community strings. Users with read-only access or who do not have access to management functions in the CLI cannot display the strings. For users with read-write access, the strings are encrypted in the CLI but are shown in the clear in the Web Management Interface.

To display the community strings in the CLI, first use the **enable password-display** command and then use the **show snmp server** command. This will display both the read-only and read-write community strings in the clear.

Adding an SNMP community string

By default, the string is encrypted. To add a community string, enter commands such as the following.

```
NetIron(config)# snmp-server community private rw
```

The command adds the read-write SNMP community string “private”.

Syntax: [no] **snmp-server community** <*string*>
ro | rw [**view** <*viewname*>] [<*standard-acl-name*> | <*standard-acl-id*> | **ipv6** <*ipv6-acl-name*>]

The <*string*> parameter specifies the community string name. The string can be up to 32 characters long.

The system modifies the configuration to `session 1.1.1.1 key 2 $XkBTb24tb0RuXA==`

By default, the community string is encrypted. If you want the community string to be in clear text, insert a **0** between **community** and <*string*>. For example,

```
NetIron(config)# snmp-server community 0 nightadmin rw
```

The software adds a prefix to the community string in the configuration. For example, the following portion of the code has the encrypted code “2”.

```
snmp-server community 2 $D?@d=8 rw
```

The prefix can be one of the following:

- 0 = the community string is not encrypted and is in clear text
- 2 = the community string uses proprietary base64 crytographic 2-way algorithm.
- The **ro | rw** parameter specifies whether the string is read-only (**ro**) or read-write (**rw**).

The **view** <*viewstring*> parameter is optional. It allows you to associate a view to the members of this community string. Enter up to 32 alphanumeric characters. If no view is specified, access to the full MIB is granted. The view that you want must exist before you can associate it to a community string. Here is an example of how to use the view parameter in the community string command.

```
NetIron(config)# snmp-s community myread ro view sysview
```

The command in this example associates the view “sysview” to the community string named “myread”. The community string has read-only access to “sysview”. For information on how create views, refer to the section “Defining SNMP views” on page 2113.

The <*standard-acl-name*> | <*standard-acl-id*> | **ipv6** <*ipv6-acl-name*> parameter is optional. It allows you to specify which ACL is used to filter the incoming SNMP packets. You can enter either the ACL name or its ID for an IPv4 ACL; for an IPv6 ACL, you must enter the keyword **ipv6** followed by the name of the IPv6 ACL. Here are examples.

```
NetIron(config) # snmp-s community myread ro view sysview 2
NetIron(config) # snmp-s community myread ro view sysview myacl
```

The command in the first example specifies that ACL group 2 filters incoming SNMP packets, whereas the command in the second example uses the IPv4 ACL group called “myacl” to filter incoming packets.

Displaying the SNMP community strings

To display the community strings in the CLI, first use the **enable password-display** command and then use the **show snmp server** command. This will display both the read-only and read-write community strings in the clear.

To display the configured community strings, enter the following command at any CLI level.

```
NetIron(config)# show snmp server
```

Syntax: show snmp server

NOTE
If display of the strings is encrypted, the strings are not displayed. Encryption is enabled by default.

Using the User-Based Security model

SNMP version 3 (RFC 2570 through 2575) introduces a User-Based Security model (RFC 2574) for authentication and privacy services.

SNMP version 1 and version 2 use community strings to authenticate SNMP access to management modules. This method can still be used for authentication. In SNMP version 3, the User-Based Security model of SNMP can be used to secure against the following threats:

- Modification of information
- Masquerading the identity of an authorized entity
- Message stream modification
- Disclosure of information

Furthermore, SNMP version 3 supports View-Based Access Control Mechanism (RFC 2575) to control access at the PDU level. It defines mechanisms for determining whether or not access to a managed object in a local MIB by a remote principal should be allowed. (Refer to the section "Defining SNMP views" on page 2113.)

Configuring your NMS

To be able to use the SNMP version 3 features.

1. Make sure that your Network Manager System (NMS) supports SNMP version 3.
2. Configure your NMS agent with the necessary users.
3. Configure the SNMP version 3 features in the PowerConnect.

Configuring SNMP version 3 on the PowerConnect

To configure SNMP version 3 on the PowerConnect, perform the tasks listed below.

1. Enter an engine ID for the management module using the **snmp-server engineid** command if you will not use the default engine ID. Refer to "Defining the engine ID" on page 2108.
2. Create views that will be assigned to SNMP user groups using the **snmp-server view** command. Refer to the "Defining SNMP views" on page 2113 for details.

3. Create ACL groups that will be assigned to SNMP user groups using the **access-list** command. Refer to 21, "Access Control List" for details.
4. Create user groups using the **snmp-server group** command. Refer to "Defining an SNMP group" on page 2109.
5. Create user accounts and associate these accounts to user groups using the **snmp-server user** command. Refer to "Defining an SNMP user account" on page 2110.

If SNMP version 3 is not configured, then community strings by default are used to authenticate access.

Even if SNMP version 3 users are configured on the device, the system will still accept SNMP version 1, 2c and 3 PDUs from the remote manager.

Defining the engine ID

A default engine ID is generated during system start up.The format of the default engine ID is derived from RFC 2571 (Architecture for SNMP frameworks) within the MIB description for object SnmpEngineID.

To determine what the default engine ID of the device is, enter the **show snmp engineid** command and find the following line.

```
Local SNMP Engine ID: 800007c70300e05290ab60
```

Refer to the section "Displaying the engine ID" on page 2111 for details.

The default engine ID guarantees the uniqueness of the engine ID for SNMP version 3. If you want to change the default engine ID, enter a command such as the following.

```
NetIron(config)# snmp-server engineid local 800007c70300e05290ab60
```

Syntax: [no] snmp-server engineid local <*hex-string*>

The **local** parameter indicates that engine ID to be entered is the ID of this device, representing an SNMP management entity.

NOTE
Since the current implementation of SNMP version 3 does not support Notification, remote engine IDs cannot be configured at this time.

The <*hex-string*> variable consists of 11 octets, entered as hexadecimal values. Each octet has two hexadecimal characters. The engine ID should contain an even number of hexadecimal characters.

The default engine ID has a maximum of 11 octets:

- Octets 1 through 4 represent the agent's SNMP management private enterprise number as assigned by the Internet Assigned Numbers Authority (IANA). The most significant bit of Octet 1 is "1". For example, "000007c7" is the ID for Brocade in hexadecimal. With Octet 1 always equal to "1", the first four octets in the default engine ID is always "800007c7" (which is 1991 in decimal).
- Octet 5 is always 03 in hexadecimal and indicates that the next set of values represent a MAC address.
- Octets 6 through 11 form the MAC address of the lowest port in the management module.

NOTE
Engine ID must be a unique number among the various SNMP engines in the management domain. Using the default engine ID ensures the uniqueness of the numbers.

Defining an SNMP group

SNMP groups map SNMP users to SNMP views. For each SNMP group, you can configure a read view, a write view, or both. Users who are mapped to a group will use its views for access control.

To configure an SNMP user group, enter a command such as the following.

```
NetIron(config)# snmp-server group admin v3 auth read all write all
```

Syntax: [no] **snmp-server group** <*groupname*>
v1 | v2c | v3
auth | noauth | priv
[**access** <*standard-acl-id*>] [**read** <*viewstring*>] [**write** <*viewstring*>] [**notify** <*viewname*>]

NOTE
The **snmp-server group** command has been enhanced to include the [notify <*viewname*>] keyword.

NOTE
This command is not used for SNMP version 1 and SNMP version 2. In these versions, groups and group views are created internally using community strings. (Refer to "Establishing SNMP community strings" on page 2105.) When a community string is created, two groups are created, based on the community string name. One group is for SNMP version 1 packets, while the other is for SNMP version 2 packets.

The **group** <*groupname*> parameter defines the name of the SNMP group to be created.

The **v1**, **v2c**, or **v3** parameter indicates which version of SNMP is used. In most cases, you will be using v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

The **auth** | **noauth** parameter determines whether authentication is required for accessing the supported views. If auth is selected, then only authenticated packets are allowed to access the view specified for the user group. Selecting **noauth** means that no authentication is required to access the specified view. Selecting **priv** means that an authentication password is required from the users.

The auth | noauth | priv parameter is available when you select v3, not v1 or v2.

The **access** <*standard-acl-id*> parameter is optional. It allows incoming SNMP packets to be filtered based on the standard ACL attached to the group.

The **read** <*viewstring*> | **write** <*viewstring*> parameter is optional. It indicates that users who belong to this group have either read or write access to the MIB.

The **notify** <*viewname*> parameter is optional. It allows trap notifications to be encrypted and sent to target hosts.

The <*viewstring*> variable is the name of the view to which the SNMP group members have access. If no view is specified, then the group has no access to the MIB.

The value of <*viewstring*> is defined by using the **snmp-server view** command. The SNMP agent comes with the "all" view, the default view that provides access to the entire MIB; however, it must be specified when creating the group. The "all" view also lets SNMP version 3 be backwards compatibility with SNMP version 1 and version 2.

NOTE
If you plan to use a view other than the "all" view, that view must have been configured before you create the user group. Refer to "Defining SNMP views" on page 2113, for details on the **include | exclude** parameters.

Defining an SNMP user account

The **snmp-server user** command does the following:

- Creates an SNMP user.
- Defines the group to which the user will be associated.
- Defines the type of authentication to be used for SNMP access by this user.

Here is an example of how to create the account.

```
NetIron(config)# snmp-s user bob admin v3 access 2 auth md5 bobmd5 priv des bobdes
```

The CLI for creating SNMP version 3 users has been updated as follows.

Syntax: **[no] snmp-server user** <*name*> <*groupname*> **v3**
[[access <*standard-acl-id*>**]**
[[encrypted] auth md5 <*md5-password*> **| sha** <*sha-password*>
[priv [encrypted] des <*des-password-key*> **| aes** <*aes-password-key*>**]]]**

The <*name*> parameter defines the SNMP user name or security name used to access the management module.

The <*groupname*> parameter identifies the SNMP group to which this user is associated or mapped. All users must be mapped to an SNMP group. Groups are defined using the **snmp-server group** command.

NOTE
The SNMP group to which the user account will be mapped should be configured before creating the user accounts; otherwise, the group will be created without any views. Also, ACL groups must be configured before configuring user accounts.

The **v3** parameter is required.

The **access** <*standard-acl-id*> parameter is optional. It indicates that incoming SNMP packets are filtered based on the ACL attached to the user account.

NOTE
The ACL specified in a user account overrides the ACL assigned to the group to which the user is mapped. If no ACL is entered for the user account, the ACL configured for the group is used to filter packets.

The encrypted parameter means that the MD5 or SHA password will be a digest value. MD5 has 16 octets in the digest. SHA has 20. The digest string has to be entered as a hexadecimal string. In this case, the agent need not generate any explicit digest. If the encrypted parameter is not used, the user is expected to enter the authentication password string for MD5 or SHA. The agent converts the password string to a digest, as described in RFC 3414.

The optional **auth md5 | sha** parameter defines the type of encryption the user must have to be authenticated. The choices are MD5 and SHA encryption (the two authentication protocols used in SNMP version 3).

The <*md5-password*> and <*sha-password*> define the password the user must use to be authenticated. These password must have a minimum of 8 characters. If the encrypted parameter is used, then the digest has 16 octets for MD5 or 20 octets for SHA.

NOTE
Once a password string is entered, the generated configuration displays the digest (for security reasons), not the actual password.

The **priv** [encrypted] parameter is optional after you enter the md5 or sha password. The **priv** parameter specifies the encryption that is used to encrypt the privacy password. If the **encrypted** keyword is used, do the following:

- If DES is the privacy protocol to be used, enter **des** <*des-password-key*> and enter a 16-octet DES key in hexadecimal format for the <*des-password-key*>. If you include the **encrypted** keyword, enter a password string of at least 8 characters.
- If AES is the privacy protocol to be used, enter **aes** and an <*aes-password-key*>. Enter either 12 (for a small key) or 16 (for a big key) characters for the <*aes-password-key*>. If you include the **encrypted** keyword, enter a password string containing 32 hexadecimal characters.

Displaying the engine ID

To display the engine ID of a management module, enter a command such as the following.

```
NetIron(config)# show snmp engineid
Local SNMP Engine ID: 800007c70300e05290ab60
Engine Boots: 3
Engine time: 5
```

Syntax: show snmp engineid

The engine ID identifies the source or destination of the packet.

The engine boots represents the number of times that the SNMP engine reinitialized itself with the same engine ID. If the engineID is modified, the boot count is reset to 0.

The engine time represents the current time with the SNMP agent.

Displaying SNMP groups

To display the definition of an SNMP group, enter a command such as the following.

```
NetIron(config)# show snmp group
groupname = exceptifgrp
security model = v3
security level = authNoPriv
ACL id = 2
readview = exceptif
writeview = <none>
```

Syntax: show snmp group

The value for security level can be one of the following.

Security level	Authentication
<none>	If the security model shows v1 or v2, then security level is blank. User names are not used to authenticate users; community strings are used instead.
noauthNoPriv	Displays if the security model shows v3 and user authentication is by user name only.
authNoPriv	Displays if the security model shows v3 and user authentication is by user name and the MD5 or SHA algorithm.
authPriv	Authentication uses MD5 or SHA. Encryption uses DES and AES protocol.

Displaying user information

To display the definition of an SNMP user account, enter a command such as the following.

```
NetIron(config)# show snmp user
username = bob
acl id = 0
group = bobgroup
security model = v3
group acl id = 0
authtype = md5
authkey = ad172674ebc09cd9448c8276db0d12f8
privtype = aes
privkey = 3c154b47996534b22b22758e23f9a71a
engine ID= 800007c703000cdbf48a00
```

Syntax: show snmp user

Interpreting varbinds in report packets

If an SNMP version 3 request packet is to be rejected by an SNMP agent, the agent sends a report packet that contains one or more varbinds. The varbinds contain additional information, showing the cause of failures. An SNMP manager application decodes the description from the varbind. The following table presents a list of varbinds supported by the SNMP agent.

Varbind object identifier	Description
1. 3. 6. 1. 6. 3. 11. 2. 1. 3. 0	Unknown packet data unit.
1. 3. 6. 1. 6. 3. 12. 1. 5. 0	The value of the varbind shows the engine ID that needs to be used in the snmp-server engineid command

Varbind object identifier	Description
1. 3. 6. 1. 6. 3. 15. 1. 1. 1. 0	Unsupported security level.
1. 3. 6. 1. 6. 3. 15. 1. 1. 2. 0	Not in time packet.
1. 3. 6. 1. 6. 3. 15. 1. 1. 3. 0	Unknown user name. This varbind can also be generated if either the: • Configured ACL for the user filters out the packet. • Group associated with the user is unknown.
1. 3. 6. 1. 6. 3. 15. 1. 1. 4. 0	Unknown engine ID. The value of this varbind would be the correct authoritative engineID that should be used.
1. 3. 6. 1. 6. 3. 15. 1. 1. 5. 0	Wrong digest.
1. 3. 6. 1. 6. 3. 15. 1. 1. 6. 0	Decryption error.

Defining SNMP views

SNMP views are named groups of MIB objects that can be associated with user accounts to allow limited access for viewing and modification of SNMP statistics and system configuration. SNMP views can also be used with other commands that take SNMP views as an argument. SNMP views reference MIB objects using object names, numbers, wildcards, or a combination of the three. The numbers represent the hierarchical location of the object in the MIB tree. You can reference individual objects in the MIB tree or a subset of objects from the MIB tree.

You can create up to 10 views on the PowerConnect. This number cannot be changed.

To create an SNMP view, enter one of the following commands.

```
NetIron(config)# snmp-server view Maynes system included
NetIron(config)# snmp-server view Maynes system.2 excluded
NetIron(config)# snmp-server view Maynes 2.3.*.6 included
NetIron(config)# write mem
```

NOTE
The **snmp-server view** command supports the MIB objects as defined in RFC 1445.

Syntax: [no] snmp-server view <*name*> <*mib_tree*> **included | excluded**

The <*name*> parameter can be any alphanumeric name you choose to identify the view. The names cannot contain spaces.

The <*mib_tree*> parameter is the name of the MIB object or family. MIB objects and MIB sub-trees can be identified by a name or by the numbers called Object Identifiers (OIDs) that represent the position of the object or sub-tree in the MIB hierarchy. You can use a wildcard (*) in the numbers to specify a sub-tree family.

The **included | excluded** parameter specifies whether the MIB objects identified by the <*mib_family*> parameter are included in the view or excluded from the view.

NOTE
All MIB objects are automatically excluded from any view unless they are explicitly included; therefore, when creating views using the **snmp-server view** command, indicate which portion of the MIB you want users to access.
For example, you may want to assign the view called “admin” a community string or user group. The “admin” view will allow access to the IronWare MIBs objects that begin with the 1.3.6.1.4.1.1991

object identifier. Enter the following command.
NetIron(config)# snmp-server view admin 1.3.6.1.4.1.1991 included
You can exclude portions of the MIB within an inclusion scope. For example, if you want to exclude the snAgentSys objects, which begin with 1.3.6.1.4.1.1991.1.1.2 object identifier from the admin view, enter a second command such as the following.
NetIron(config)# snmp-server view admin 1.3.6.1.4.1.1991.1.1.2 excluded
Note that the exclusion is within the scope of the inclusion.

To delete a view, use the no parameter before the command.

SNMP v3 configuration examples

The examples below shows how to configure SNMP v3.

Simple SNMP v3 configuration

```
NetIron(config)#snmp-s group admingrp v3 priv read all write all notify all
NetIron(config)#snmp-s user adminuser admingrp v3 auth md5 <auth password> priv
<privacy password>
NetIron(config)#snmp-s host <dest-ip> adminuser
```

More detailed SNMP v3 configuration

```
NetIron(config)#snmp-server view internet internet included
NetIron(config)#snmp-server view system system included
NetIron(config)#snmp-server community ..... ro
NetIron(config)#snmp-server community ..... rw
NetIron(config)#snmp-server contact isc-operations
NetIron(config)#snmp-server location sdh-pillbox
NetIron(config)#snmp-server host 128.91.255.32 .....
NetIron(config)#snmp-server group ops v3 priv read internet write system
NetIron(config)#snmp-server group admin v3 priv read internet write internet
NetIron(config)#snmp-server group restricted v3 priv read internet
NetIron(config)#snmp-server user ops ops v3 encrypted auth md5
ab8e9cd6d46e7a270b8c9549d92a069 priv encrypted des
0e1b153303b6188089411447dbc32de
NetIron(config)#snmp-server user admin admin v3 encrypted auth md5
0d8a2123f91bfbd8695fef16a6f4207b priv encrypted des
18e0cf359fce4fcd60df19c2b6515448
NetIron(config)#snmp-server user restricted restricted v3 encrypted auth md5
261fd8f56a3ad51c8bcec1e4609f54dc priv encrypted des
d32e66152f89de9b2e0cb17a65595f43
```

Chapter

Remote Network Monitoring 55

The following Remote Network Monitoring features are supported by PowerConnect B-MLXe Series.

- Remote Network Monitoring
- Statistics (RMON Group 1)
- History (RMON Group 2)
- Alarms (RMON Group 3)
- Events (RMON Group 9)

This chapter describes the remote monitoring features available on PowerConnect products:

- **Remote Monitoring (RMON) statistics** – All PowerConnect products support RMON statistics on the individual port level. Refer to "RMON support" on page 2116.
- **sFlow** – sFlow collects interface statistics and traffic samples from individual interfaces on a PowerConnect and exports the information to a monitoring server. Refer to Chapter 57, "sFlow".

Basic management

The following sections contain procedures for basic system management tasks.

Viewing system information

You can access software and hardware specifics for a PowerConnect.

To view the software and hardware details for the system, enter the **show version** command.

```
NetIron# show version
```

Syntax: show version

Viewing configuration information

You can view a variety of configuration details and statistics with the show option. The **show** option provides a convenient way to check configuration changes before saving them to flash.

The show options available will vary for the PowerConnect and by configuration level.

To determine the available show commands for the system or a specific level of the CLI, enter the following command.

```
NetIron# show ?
```

Syntax: show <*option*>

You also can enter "show" at the command prompt, then press the TAB key.

Viewing port statistics

Port statistics are polled by default every 10 seconds.

You can view statistics for ports by entering the following **show** commands:

- show interfaces
- show configuration

Viewing STP statistics

You can view a summary of STP statistics for the PowerConnect. STP statistics are by default polled every 10 seconds.

To view spanning tree statistics, enter the **show span** command. To view STP statistics for a VLAN, enter the **span vlan** command.

Clearing statistics

You can clear statistics for many parameters with the clear option.

To determine the available **clear** commands for the system, enter the following command.

```
NetIron# clear ?
```

Syntax: clear <*option*>

You also can enter “clear” at the command prompt, then press the TAB key.

NOTE
Clear commands are found at the Privileged EXEC level.

RMON support

The RMON agent supports the following groups. The group numbers come from the RMON specification (RFC 1757):

- Statistics (RMON Group 1)
- History (RMON Group 2)
- Alarms (RMON Group 3)
- Events (RMON Group 9)

The CLI allows you to make configuration changes to the control data for these groups, but you need a separate RMON application to view and display the data graphically.

Statistics (RMON group 1)

Count information on multicast and broadcast packets, total packets sent, undersized and oversized packets, CRC alignment errors, jabbers, collision, fragments and dropped events is collected for each port on a PowerConnect.

No configuration is required to activate collection of statistics for the PowerConnect. This activity is by default automatically activated at system start-up.

NOTE
The PowerConnect system provides limited MIB counters. Dell uses "rmon_giant" to represent oversized packet, i.e 9216 and above.

You can view a textual summary of the statistics for all ports by entering the following CLI command.

```
NetIron(config)# show rmon statistics
Ethernet statistics 1 is active, owned by monitor
 Interface 1/1 (ifIndex 1) counters
                  Octets          0
              Drop events         0                  Packets          0
           Broadcast pkts         0            Multicast pkts         0
     CRC alignment errors         0            Undersize pkts         0
            Oversize pkts         0                 Fragments         0
                  Jabbers         0               Collisions         0
           64 octets pkts         0     65 to 127 octets pkts         0
   128 to 255 octets pkts         0    256 to 511 octets pkts         0
  512 to 1023 octets pkts         0  1024 to 1518 octets pkts         0
```

Syntax: show rmon statistics [<*num*> | **ethernet** <*slot/port*> | **management** <*num*> | | **begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

The <*portnum*> parameter specifies the port number. You can use the physical port number or the SNMP port number. The physical port number is based on the product.

- The ports are numbered according to slot and port. For example, the first port in slot 1 is 1/1. The third port in slot 7 is 7/3.

The SNMP numbers of the ports start at 1 and increase sequentially. For example, if you are using a Chassis device and slot 1 contains an 8-port module, the SNMP number of the first port in slot 2 is 9. The physical port number of the same port is 2/1.

This command shows the following information.

TABLE 406 Export configuration and statistics

This line...	Displays...
Octets	The total number of octets of data received on the network. This number includes octets in bad packets. This number does not include framing bits but does include Frame Check Sequence (FCS) octets.
Drop events	Indicates an overrun at the port. The port logic could not receive the traffic at full line rate and had to drop some packets as a result. The counter indicates the total number of events in which packets were dropped by the RMON probe due to lack of resources. This number is not necessarily the number of packets dropped, but is the number of times an overrun condition has been detected.
Packets	The total number of packets received. This number includes bad packets, broadcast packets, and multicast packets.
Broadcast pkts	The total number of good packets received that were directed to the broadcast address. This number does not include multicast packets.

TABLE 406 Export configuration and statistics (Continued)

This line...	Displays...
Multicast pkts	The total number of good packets received that were directed to a multicast address. This number does not include packets directed to the broadcast address.
CRC alignment errors	The total number of packets received that were from 64 – 1518 octets long, but had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). The packet length does not include framing bits but does include FCS octets.
Undersize pkts	The total number of packets received that were less than 64 octets long and were otherwise well formed. This number does not include framing bits but does include FCS octets.
Fragments	The total number of packets received that were less than 64 octets long and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). It is normal for this counter to increment, since it counts both runts (which are normal occurrences due to collisions) and noise hits. This number does not include framing bits but does include FCS octets.
Oversize packets	The total number of packets received that were longer than 1518 octets and were otherwise well formed. This number does not include framing bits but does include FCS octets.
Jabbers	The total number of packets received that were longer than 1518 octets and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). **NOTE:** This definition of jabber is different from the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). These documents define jabber as the condition where any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms. This number does not include framing bits but does include FCS octets.
Collisions	The best estimate of the total number of collisions on this Ethernet segment.
64 octets pkts	The total number of packets received that were 64 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
65 to 127 octets pkts	The total number of packets received that were 65 – 127 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
128 to 255 octets pkts	The total number of packets received that were 128 – 255 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
256 to 511 octets pkts	The total number of packets received that were 256 – 511 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
512 to 1023 octets pkts	The total number of packets received that were 512 – 1023 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
1024 to 1518 octets pkts	The total number of packets received that were 1024 – 1518 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.

NOTE
The number of entries in a RMON statistics table directly corresponds to the number of ports on a system. For example, if the system is a 26 port device, there will be 26 entries in the statistics display.

History (RMON group 2)

All active ports by default will generate two history control data entries per active PowerConnect interface. An active port is defined as one with a link up. If the link goes down the two entries are automatically be deleted.

Two history entries are generated for each device:

- a sampling of statistics every 30 seconds
- a sampling of statistics every 30 minutes

The history data can be accessed and displayed using any of the popular RMON applications

A sample RMON history command and its syntax is shown below.

```
NetIron(config)# rmon history 1 interface 1 buckets 10 interval 10 owner nyc02
```

Syntax: **rmon history** <*entry-number*> **interface ethernet** <*slot/port*> | **management** <*num*> **buckets** <*number*> **interval** <*sampling-interval*> **owner** <*text-string*>

You can modify the sampling interval and the bucket (number of entries saved before overwrite) using the CLI. In the above example, owner refers to the RMON station that will request the information.

NOTE
To review the control data entry for each port or interface, enter the **show rmon history** command.

Alarm (RMON group 3)

Alarm is designed to monitor configured thresholds for any SNMP integer, time tick, gauge or counter MIB object. Using the CLI, you can define what MIB objects are monitored, the type of thresholds that are monitored (falling, rising or both), the value of those thresholds, and the sample type (absolute or delta).

An alarm event is reported each time that a threshold is exceeded. The alarm entry also indicates the action (event) to be taken if the threshold be exceeded.

A sample CLI alarm entry and its syntax is shown below.

```
NetIron(config)# rmon alarm 1 ifInOctets.6 10 delta rising-threshold 100 1 falling
threshold 50 1 owner nyc02
```

Syntax: **rmon alarm** <*entry-number*> <*MIB-object.interface-num*> <*sampling-time*> <*sample-type*>
<*threshold-type*> <*threshold-value*> <*event-number*> <*threshold-type*> <*threshold-value*> <*event-number*>
owner <*text-string*>

The <*sample-type*> can be absolute or delta.

The <*threshold-type*> can be falling-threshold or rising-threshold.

Event (RMON group 9)

There are two elements to the Event Group—the ***event control table*** and the ***event log table***.

The event control table defines the action to be taken when an alarm is reported. Defined events can be found by entering the CLI command, show event. The Event Log Table collects and stores reported events for retrieval by an RMON application.

A sample entry and syntax of the event control table is shown below.

```
NetIron(config)# rmon event 1 description ‘testing a longer string’ log-and-trap
public owner nyc02
```

Syntax: **rmon event** <*event-entry*> **description** <*text-string*> **log | trap | log-and-trap | owner** <*rmon-station*>

Chapter

Configuring Management VRF 56

The following management Virtual Routing and Forwarding (VRF) features are supported by the NetIron MLX Series devices:

- IPv4 management VRF
- IPv6 management VRF

Management VRF overview

The management VRF is used to provide secure management access to the device by sending inbound and outbound management traffic through the VRF specified as a global management VRF and through the out-of-band management port, thereby isolating management traffic from the network data traffic.

By default, the inbound traffic is unaware of VRF and allows incoming packets from any VRF, including the default VRF. The outbound traffic is only through the default VRF. The default VRF consists of out-of-band management port and all the LP ports that do not belong to any other VRFs.

Any VRF, except the default VRF, can be configured as a management VRF. When a management VRF is configured, the management traffic is allowed through the ports belonging to the specified VRF and the out-of-band management port. The management traffic through the ports belonging to the other VRFs and the default VRF are dropped and the rejection statistics are incremented.

If the management VRF is not configured, the management applications will follow the default behavior. The management VRF configuration is applicable for both IPv4 and IPv6 management traffic.

The management VRF is supported by the following management applications:

- SNMP server
- SNMP trap generator
- Telnet server
- SSH server
- Telnet client
- RADIUS client
- TACACS+ client
- TFTP
- SCP
- Syslog

NOTE
The management VRF is not applicable to inbound and outbound traffic of the **ping** and **traceroute** commands. These commands use the VRF specified in the command or the default VRF, if no VRF is specified.

Source interface and management VRF compatibility

There is a source interface configuration associated with the management applications. When a source interface is configured, the management applications use the lowest configured IP address of the specified interface as source IP address in all the outgoing packets. If the configured interface is not part of the management VRF, the response packet will not reach the destination. If the compatibility check fails while configuring either the management VRF or the source interface, the following warning message will be displayed. However, the configuration command will be accepted.

```
The source-interface for Telnet, TFTP is not part of the management-vrf
```

Supported management applications

This section explains the management VRF support provided by the management applications.

SNMP server

When the management VRF is configured, the SNMP server receives SNMP requests and sends SNMP responses only through the ports belonging to the management VRF and through the out-of-band management port.

Any change in the management VRF configuration becomes immediately effective for the SNMP server.

SNMP trap generator

When the management VRF is configured, the SNMP trap generator sends traps to trap hosts through the ports belonging to the management VRF and through the out-of-band management port.

Any change in the management VRF configuration becomes immediately effective for the SNMP trap generator.

NOTE
The SNMP source interface configuration command **snmp-server trap-source** must be compatible with the management VRF configuration. Refer to "Source interface and management VRF compatibility" on page 2122.

Telnet server

When the management VRF is configured, the incoming Telnet connection requests are allowed only from the ports belonging to the management VRF and from the out-of-band management port. Management VRF enforcement is only done during the establishment of a connection. Once the connection is established, no further management VRF enforcement is done.

To allow the incoming Telnet connection requests only from the management VRF and not from the out-of-band management port, enter the following command.

```
NetIron(config)# telnet strict-management-vrf
```

The previous command is applicable only when the management VRF is configured. If not, the command issues the following warning message.

```
Warning - Management-vrf is not configured.
```

For the Telnet server, changes in the management VRF configuration or configuring the **telnet strict-management-vrf** command will not affect the existing Telnet connections and the changes will be applied only to the new incoming connection requests.

SSH server

When the management VRF is configured, the incoming SSH connection requests are allowed only from the ports belonging to the management VRF and from the out-of-band management port. Management VRF enforcement is only done during the establishment of a connection. Once the connection is established, no further management VRF enforcement is done.

To allow the incoming SSH connection requests only from the management VRF and not from the out-of-band management port, enter the following command.

```
NetIron(config)# ip ssh strict-management-vrf
```

The previous command is applicable only when the management VRF is configured. If not, the command issues the following warning message.

```
Warning - Management-vrf is not configured.
```

For the SSH server, changes in the management VRF configuration or configuring the **ip ssh strict-management-vrf** command will not affect the existing SSH connections and the changes will be applied only to the new incoming connection requests.

Telnet client

When the VRF name is specified in the **telnet vrf** command, the Telnet client initiates Telnet requests only from the ports belonging to the specified VRF.

To configure the VRF name in outbound Telnet sessions, enter the following command.

```
NetIron(config)# telnet vrf red 209.157.22.39
```

Syntax: telnet vrf <*vrf-name*> <*IPv4 address*> | **ipv6** <*IPv6 address*>

The <*vrf-name*> variable specifies the name of the pre-configured VRF.

RADIUS client

When the management VRF is configured, the RADIUS client will sends RADIUS requests or receives responses only through the ports belonging to the management VRF and through the out-of-band management port.

Any change in the management VRF configuration will be immediately effective for the RADIUS client.

NOTE
The RADIUS source interface configuration command **ip radius source-interface** must be compatible with the management VRF configuration. Refer to "Source interface and management VRF compatibility" on page 2122.

TACACS+ client

When the management VRF is configured, the TACACS+ client establishes connections with TACACS+ servers only through the ports belonging to the management VRF and the out-of-band management port.

For the TACACS+ client, any change in the management VRF configuration will not affect the existing TACACS+ connections and the changes will be applied only to the new TACACS+ connections.

NOTE
The TACACS+ source interface configuration command **ip tacacs source-interface** must be compatible with the management VRF configuration. Refer to "Source interface and management VRF compatibility" on page 2122.

TFTP

When the management VRF is configured, TFTP will send or receive the data and acknowledgements only through the ports belonging to the management VRF and through the out-of-band management port.

Any change in the management VRF configuration will be immediately effective for TFTP. You cannot change in the management VRF configuration while TFTP is in progress.

NOTE
The TFTP source interface configuration command **ip tftp source-interface** must be compatible with the management VRF configuration. Refer to "Source interface and management VRF compatibility" on page 2122.

SCP

SCP uses SSH as underlying transport. The behavior of SCP is similar to the SSH server. For more information, refer to "SSH server" on page 2123.

Syslog

When the management VRF is configured, the Syslog module sends log messages only through the ports belonging to the management VRF and the out-of-band management port.

Any change in the management VRF configuration will be immediately effective for Syslog.

NOTE
The Syslog source interface configuration command **ip syslog source-interface** must be compatible with the management VRF configuration. Refer to "Source interface and management VRF compatibility" on page 2122.

Configuring a global management VRF

To configure a VRF as a global management VRF, enter the following command.

```
NetIron(config)# management-vrf mvrf
```

Syntax: [no] management-vrf <*vrf-name*>

The <*vrf-name*> parameter specifies the name of the pre-configured VRF. If the VRF is not pre-configured, the command execution fails and displays the following error message.

```
Error - VRF <vrf-name> doesn't exist
```

When the management VRF is configured, the software generates the following Syslog message.

```
SYSLOG: VRF <vrf-name> has been configured as management-vrf
```

Enter the **no** form of the command to remove the management VRF. When the management VRF is deleted, the software generates the following Syslog message.

```
SYSLOG: VRF <vrf-name> has been un-configured as management-vrf
```

Configuration notes

Consider the following configuration notes:

- If there is a management VRF already configured, you must remove the existing management VRF configuration before configuring a new one. If not, the system displays the following error message.

  ```
  NetIron(config)# management-vrf  red
  Error - VRF mvrf already configured as management-vrf
  ```

- If you try to delete a management VRF that was not configured, the system displays the following error message.

  ```
  NetIron(config)# no management-vrf red
  Error - VRF red is not the current management-vrf
  ```

- The deletion or modification of the VRF will fail if the specified VRF is currently configured as the management VRF. Attempting to do so causes the system to return the following error message.

  ```
  NetIron(config)# no ip vrf mvrf
  Error - Cannot modify/delete a VRF which is configured as management-vrf
  ```

Displaying the management VRF information

To display IP Information for a specified VRF, enter the following command at any level of the CLI.

```
NetIron(config)# show vrf mvrf
VRF mvrf, default RD 1:1, Table ID 1 IFL ID 131071
Label: 500000,  Label-Switched Mode: OFF
Configured as management-vrf
IP Router-Id: 2.2.2.2
  Interfaces:
    e2/2
  No Export VPN route-target communities
  No Import VPN route-target communities
```

```
No import route-map
  No export route-map
  Address Family IPv4
    Max Routes: 5120
    No Export VPN route-target communities
    No Import VPN route-target communities
 Address Family IPv6
    Max Routes: 128
    No Export VPN route-target communities
    No Import VPN route-target communities
```

Syntax: show vrf <*vrf-name*>

The <*vrf-name*> parameter specifies the VRF for which you want to display IP information.

Table 407 displays a description of the output from the **show vrf** command.

TABLE 407 Output from the **show vrf** command

This field...	Displays...
VRF <*name*>	The name of the VRF.
default RD	The default route distinguisher for the VRF.
Table ID	The table ID for the VRF.
IFL ID	The Internal Forwarding Lookup Identifier (IFL-ID) for ports in the VRF instance.
Label	The unique VRF label that has been assigned to the specified VRF.
Label-Switched Mode	Indicates whether Label-Switched Mode is ON or OFF.
Configured as management-vrf	Indicates that the specified VRF is configured as a management VRF.
IP Router-Id	The 32-bit number that uniquely identifies the router.
import route-map	The name of the import route-map, if any, that is configured for this management VRF.
export route-map	The name of the export route-map if a route-map has been configured for this management VRF.

The **show who** command displays information about the management VRF from which the Telnet and SSH connection has been established.

```
NetIron(config)# show who
Console connections:
        established, monitor enabled, privilege super-user, in config mode
        1 minutes 47 seconds in idle
Telnet server status: Enabled
Telnet connections (inbound):
 1      established, client ip address 10.53.1.181, user is lab, privilege
super-user
        using vrf default-vrf.
        2 minutes 46 seconds in idle
 2      established, client ip address 20.20.20.2, user is lab, privilege
super-user
        using vrf mvrf.
        16 seconds in idle
 3      closed
```

```
 4      closed
 5      closed
Telnet connections (outbound):
 6      established, server ip address 20.20.20.2, from Telnet session 2, ,
privilege super-user
        using vrf mvrf.
        12 seconds in idle
 7      closed
 8      closed
 9      closed
 10     closed
SSH server status: Enabled
SSH connections:
 1      established, client ip address 10.53.1.181, privilege super-user
        using vrf default-vrf.
        you are connecting to this session
        3 seconds in idle
 2      established, client ip address 20.20.20.2, privilege super-user
        using vrf mvrf.
        48 seconds in idle
 3      closed
 4      closed
 5      closed
 6      closed
 7      closed
 8      closed
 9      closed
 10     closed
 11     closed
 12     closed
 13     closed
 14     closed
 15     closed
 16     closed
```

Syntax: show who

To display the packets and sessions rejection statistics due to failure in management VRF validation, enter the following command.

```
NetIron(config)# show management-vrf

Management VRF name : mvrf
Management Application Rx Drop Pkts Tx Drop Pkts
SNMP Engine              36         0
RADIUS Client            0          8
TFTP Client              0          4
SNMP Notifications       -          55
SysLogs                  -          78

TCP Connection rejects:
Telnet            : 1
SSH               : 1
TACACS+ Client    : 8
```

Syntax: show management-vrf

Table 408 displays a description of the output from the **show management-vrf** command.

TABLE 408 Output from the **show management-vrf** command

This field...	Displays...
Management VRF name	Displays the configured management VRF name.
Management Application	Displays the management application names.
Rx Drop Pkts	Displays the number of packets dropped in the inbound traffic.
Tx Drop Pkts	Displays the number of packets dropped in the outbound traffic.
TCP Connection rejects	Displays the number of TCP connections per application rejected due to management VRF validation.

Make sure that the management VRF is configured before executing the **show management-vrf** command. If not, the system will display the following error message.

```
Error - Management VRF is not configured.
```

To clear the management VRF rejection statistics, enter the following command.

```
NetIron(config)# clear management-vrf-stats
```

Syntax: clear management-vrf-stats

Chapter

sFlow 57

The following sFlow features are supported by PowerConnect B-MLXe Series.

- sFlow (RFC 3176)
- sFlow (v5 Support)
- sFlow Support on MPLS L2/3 VPN Endpoints
- sFlow Support on MPLS Uplinks
- ACL-based sFlow

sFlow is a system for observing traffic flow patterns and quantities within and among a set of PowerConnect devices. To support sFlow, the system:

- Samples the packet flows
- Collects the packet headers from sampled packets to gather ingress-egress information on these packets
- Composes the collected information into flow sample messages
- Relays these messages to an external device known as a collector

Participating devices also relay byte and packet counter data (counter samples) for ports to the collector.

sFlow is described in RFC 3176, “InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks”. Refer to this RFC to determine the contents of the sampled packet..

Dell also support sFlow v5 which replaces version outlined in RFC 3176.

Configuration considerations

Sample data is collected from inbound traffic on ports enabled for sFlow. However, byte and packet counters that are sent to the collector include traffic statistics for both the ingress and egress directions.The actual IP source address of the IP header is decided from the router port address of the best route to the sFlow collector IP address.

NOTE
Interface module processors directly forward sFlow packets to the specified sFlow collectors. The sFlow collector is reachable via ports on any of the Interface modules. We do not recommend that an sFlow collector be connected to a Management module’s Ethernet port.

NOTE
sFlow is implemented in the default VRF only. Therefore, sFlow data is only accessible by the sFlow collector (sFlow destination host(s)) defined in the default VRF.

Source address

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the IP address of the device that sent the data.

sFlow looks for an IP address in following order, and uses the first address found:

- The router ID configured by the **ip router-id** command
- The first IP address on the lowest-numbered loopback interface
- The first IP address on the lowest-numbered virtual interface
- The first IP address on any interface

NOTE
If an IP address in not already configured when you enable sFlow, the feature uses the source address 0.0.0.0. To display the agent_address, enable sFlow, then enter the **show sflow** command. Refer to "Enabling sFlow forwarding" on page 2134 and "Displaying sFlow information" on page 2138.

NOTE
If you change the address sFlow will use for the agent_address, you must disable and re-enable sFlow to enable the feature to use the changed address.

Sampling rate

The ***sampling rate*** is the average ratio of the number of packets incoming on an sflow enabled port, to the number of flow samples taken from those packets. PowerConnect ports send only the sampled traffic to the CPU. sFlow sampling requires high LP CPU usage, which can affect performance in some configurations especially if a high sampling rate is implemented.

Configured rate and actual rate

When you enter a sampling rate value, this value is the configured rate. The software rounds the value you enter to the next higher odd power of 2 to obtain the actual rate. This value becomes the actual sampling rate. For example, if the configured sampling rate is 1000, then the actual rate is 2048 and 1 in 2048 packets are sampled by the hardware.

Extended router information

Extended router information contains information for the next hop router. This information includes the next hop router's IP address and the outgoing VLAN ID. Extended router information also includes the source IP address prefix length and the destination IP address prefix length.

Note that in IPv4, prefix length of source and destination IP addresses is collected only if BGP is configured on the devices.

Extended gateway information

Extended gateway information is included in an sFlow sampled packet if BGP is enabled. The extended gateway information includes the following BGP information about the packet's destination route:

- This router's autonomous system (AS) number

- The route's source IP AS
- The route's source peer AS
- The AS path to the destination

In BGP configured routers AS-Path information is collected from each node traversed by the sFlow packets.

NOTE
AS communities and local preferences are not included in the sampled packets.

To obtain extended gateway information use "struct extended_gateway" as described in RFC 3176.

sFlow support for MPLS

In addition to the regular Layer 2 or Layer 3 information export supported across devices, the PowerConnect supports the exporting of MPLS or VPN information in sFlow when sFlow sampling is configured on VPN endpoint interfaces. This includes VLL, VPLS, and VRF customer endpoint interfaces. This functionality allows service providers to collect sFlow information from VPN customers.

For incoming packets to an endpoint interface sampled by sFlow, the following additional information is collected and exported in the sFlow packets:

- **MPLS VC information:** including VC name, VC index, and VC label COS
- **MPLS tunnel information:** including the LSP tunnel name, the tunnel index as assigned by the router, and the tunnel COS used

NOTE
IP over MPLS (non-L3VPN or VRF) packets are not supported for sFlow processing.

Enhancements to support MPLS in sFlow packet export format are described in "sFlow Version 5" which is available at *www.sflow.org*.

sFlow with VPLS local switching

This feature allows sFlow to carry the original VLAN ID of the incoming traffic under scenarios where a VPLS has multiple endpoints and different endpoints with different VLAN IDs – implementing automatic VLAN ID translation.

When VPLS CPU protection is enabled in conjunction with sFlow, hardware flooded unknown unicast packets will be marked with a source VLAN ID of 0. Please note that this behavior is not specific to VPLS local switching but is applicable to all VPLS traffic.

NOTE
You have to configure MAs with different MD levels to monitor the different endpoints with different VLAN IDs in the same VPLS instance.

Configuring and enabling sFlow

To configure sFlow:

- Specify collector information. The collector is the external device to which you are exporting the sFlow data. You can specify up to four collectors.
- **Optional** – Change the polling interval.
- **Optional** – Change the sampling rate.
- Enable sFlow globally.
- Enable sFlow forwarding on individual interfaces.

NOTE
If you change the router ID or other IP address value that sFlow uses for its agent_address, you need to disable and then re-enable sFlow to cause the feature to use the new source address.

Specifying the collector

sFlow exports traffic statistics to an external collector. You can specify up to four collectors. You can specify more than one collector with the same IP address if the UDP port numbers are unique. You can have up to four unique combinations of IP address and UDP port number.

NOTE
sFlow is implemented in the default VRF only. Therefore, sFlow data is only accessible by the sFlow collector (sFlow destination host(s)) defined in the default VRF.

To specify sFlow collectors, enter a command such as the following.

```
NetIron(config)# sflow destination 10.10.10.1
```

This command specifies a collector with IP address 10.10.10.1, listening for sFlow data on UDP port 6343.

Syntax: [no] sflow destination <*ip-addr*> [<*dest-udp-port*>]

The <*ip-addr*> parameter specifies the collector's IP address.

The <*dest-udp-port*> parameter specifies the UDP port on which the sFlow collector will be listening for exported sFlow data. The default port number is 6343.

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the device that sent the data. Refer to "Source address" on page 2130.

Changing the polling interval

The polling interval defines how often sFlow byte and packet counter data for a port are sent to the sFlow collectors. If multiple ports are enabled for sFlow, the PowerConnect staggers transmission of the counter data to smooth performance. For example, if sFlow is enabled on two ports and the polling interval is 20 seconds, the PowerConnect sends counter data every ten seconds. The counter data for one of the ports are sent after ten seconds, and counter data for the other port are sent after an additional ten seconds. Ten seconds later, new counter data for the first port are sent. Similarly, if sFlow is enabled on five ports and the polling interval is 20 seconds, the PowerConnect sends counter data every four seconds.

The default polling interval is 20 seconds. You can change the interval to a value from 1 to any higher value. The interval value applies to all interfaces on which sFlow is enabled. If you set the polling interval to 0, counter data sampling is disabled.

To change the polling interval, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# sflow polling-interval 30
```

Syntax: [no] sflow polling-interval <*secs*>

The <*secs*> parameter specifies the interval and can be from 1 to any higher value. The default is 20 seconds. If you specify 0, counter data sampling is disabled.

Changing the sampling rate

The sampling rate is the average ratio of the number of packets incoming on an sFlow-enabled port, to the number of flow samples taken from those packets. By default, all sFlow-enabled ports use the default sampling rate, which is 2048. With a sampling rate of 2048, on average, one in every 2048 packets forwarded on an interface is sampled.

You can change the default (global) sampling rate. You also can change the rate on an individual port, overriding the default sampling rate.

NOTE
sFlow uses CPU resources to send sFlow samples to the collector. If you set a low sampling value on a high rate interface (for example 10 GE), the Interface module CPU utilization can become high.

Configuration considerations

The sampling rate is a fraction in the form 1/N, meaning that, on average, one out of every N packets will be sampled. The **sflow sample** command at the global level or port level specifies N, the denominator of the fraction. Thus a higher number for the denominator means a lower sampling rate since fewer packets are sampled. Likewise, a lower number for the denominator means a higher sampling rate because more packets are sampled. For example, if you change the denominator from 2,000 to 512, the sampling rate increases because four times as many packets will be sampled.

NOTE
It is recommended that you do not change the denominator to a value lower than the default. Sampling requires CPU resources. Using a low denominator for the sampling rate can cause high CPU utilization.

Change to global rate

If you change the global sampling rate, the change is applied to all sFlow-enabled ports ***except*** those ports on which you have already explicitly set the sampling rate. For example, suppose that sFlow is enabled on ports 1/1, 1/2, and 5/1. If you configure the sampling rate on port 1/1 but leave the other two ports using the default rate, then a change to the global sampling rate applies to ports 1/2 and 5/1 but not port 1/1. sFlow assumes that you want to continue using the sampling rate you explicitly configured on an individual port even if you globally change the sampling rate for the other ports.

Sampling rate for new ports

When you enable sFlow on a port, the port's sampling rate is set to the global default sampling rate. This also applies to ports on which you disable and then re-enable sFlow. The port does not retain the sampling rate it had when you disabled sFlow on the port, even if you had explicitly set the sampling rate on the port.

Changing the default sampling rate

To change the default (global) sampling rate, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# sflow sample 2048
```

Syntax: [no] sflow sample <*num*>

The <*num*> parameter specifies the average number of packets from which each sample will be taken. In PowerConnect, the sampling rate you configure is the actual sampling rate. You can enter 512 – 2147483648. The default is 2048.

Changing the sampling rate on a port

You can configure an individual port to use a different sampling rate than the global default sampling rate. This is useful in cases where ports have different bandwidths. For example, if you are using sFlow on 10/100 ports and Gigabit Ethernet ports, you might want to configure the Gigabit ports to use a higher sampling rate (and thus gather fewer samples per number of packets) than the 10/100 ports.

To change the sampling rate on an individual port, enter a command such as the following at the configuration level for the port.

```
NetIron(config-if-e10000-1/1)# sflow sample 8192
```

Syntax: [no] sflow sample <*num*>

The <*num*> parameter specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. The actual sampling rate becomes one of the values listed in "Changing the default sampling rate" on page 2134.

Enabling sFlow forwarding

sFlow exports data only for the interfaces on which you enable sFlow forwarding. You can enable sFlow forwarding on Ethernet or POS interfaces

To enable sFlow forwarding:

- Globally enable the sFlow feature.
- Enable sFlow forwarding on individual interfaces.

NOTE
Before you enable sFlow, make sure the device has an IP address that sFlow can use as its source address. Refer to "Source address" on page 2130 for the source address requirements.

Enabling sFlow forwarding

To enable sFlow forwarding, enter commands such as the following.

```
NetIron(config)# sflow enable
NetIron(config)# interface ethernet 1/1 to 1/8
NetIron(config-mif-1/1-1/8)# sflow forwarding
```

These commands globally enable sFlow, then enable sFlow forwarding on Ethernet ports 1/1 – 1/8. You must use both the **sflow enable** and **sflow forwarding** commands to enable the feature.

Syntax: [no] sflow enable

Syntax: [no] sflow forwarding

NOTE
Data for POS ports is sampled using Ethernet format. PPP or HDLC header of the sampled POS packet is replaced with an Ethernet header. PPP or HDLC control packets or ISIS packets transmitted or received at a POS port are not sampled. Such packets are not included in the number of packets from which each sample is taken.

NOTE
sFlow packets cannot be forwarded from a management interface. You must configure an IP interface on an Interface module to forward sFlow packets.

ACL-based Inbound sFlow

With this release, the Multi-Service IronWare software supports using an IPv4 or IPv6 ACL to select sample traffic to be sent to an sFlow collector. The data matching an ACL clause can be collected to observe traffic flow patterns and quantities between a set of switches and routers. To accommodate collecting sFlow through standard procedures and using ACL-filtered traffic, Dell created the Proprietary Tag Type 1991 that encapsulates the sFlow samples obtained through ACL-based sFlow and separates them from the sequence flow of other sFlow samples. Figure 232 shows the format of an sFlow packet which illustrates the differences between a standard sFlow payload and an ACL-based payload.

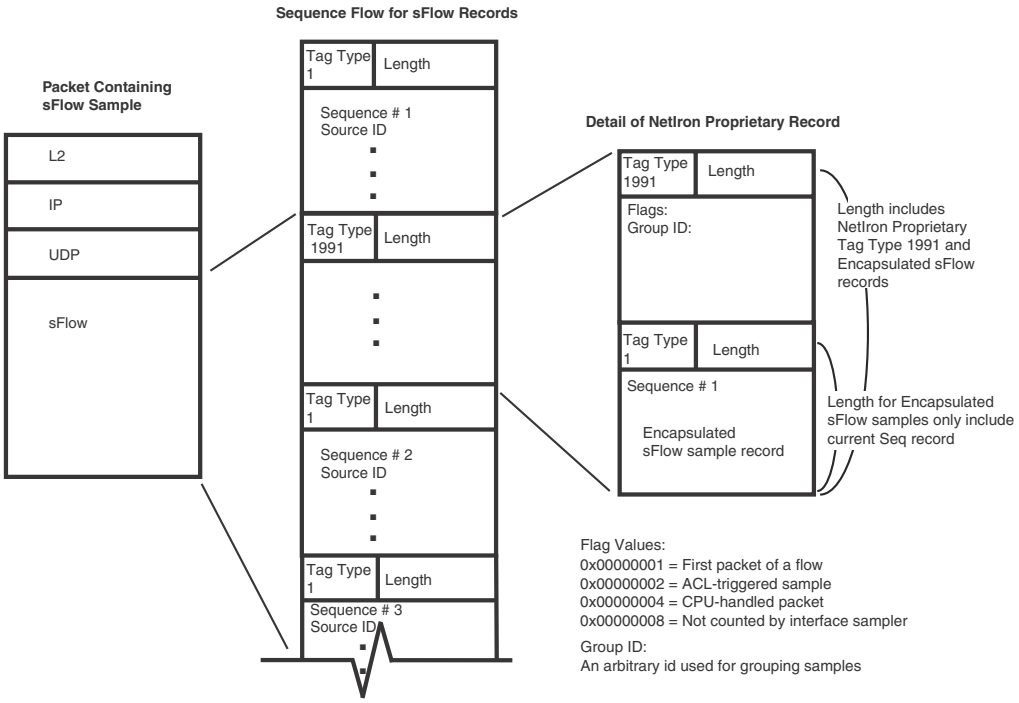
As shown in Figure 232, sFlow is carried in a UDP packet. Within the UDP packet, the sFlow contents are carried in individual samples that are identified by a Tag Type and a Length variable. The standard values for the Tag Types are: 1 = sampled packet and 2 = counter sample. The length variable describes the length of the sample. Within the sample are other variables including the Sequence number and the Source ID.

Dell has introduced the proprietary Tag Type 1991 to identify ACL-based sFlow samples. For these samples, standard Tag Type 1 samples collected using ACL-based Inbound sFlow are encapsulated in a Tag Type 1991 sample. The length variable identifies the entire length of the Tag Type 1991 sample including the encapsulated Tag Type 1 sample.The encapsulated sample has a length variable of its own that only identifies the length of that sample.

The Tag Type 1991 samples are sequenced separately from the unencapsulated Tag Type 1 samples. For instance in the packet detail described in the "Sequence Flow for sFlow Records" in Figure 232, the top sFlow record with Tag Type 1 begins with the sequence number 1. The next sFlow record is of Tag Type 1991 which indicates that the sample contained is from ACL-based sFlow. Encapsulated within this ACL-based sFlow sample is an sFlow sample record of Tag Type 1. The ACL-based sFlow sample (which contains the Type 1 sample) is followed by an unencapsulated Tag Type 1 sFlow sample. That unencapsulated Tag Type 1 sFlow sample follows the sequence numbering of the first unencapsulated Tag Type 1 sFlow sample which gives it a sequence number of 2.

This is useful in cases where an sFlow collector does not recognize Tag Type 1991. In these situations, the Tag Type 1991 samples can be ignored without disrupting the sFlow sequence numbers. It is also useful for indentifying samples obtained using ACL-based sFlow on which other processing might be performed.

FIGURE 232 sFlow packet format



Configuring ACL-based Inbound sFlow

The followings sections describe how to configure ACL-based Inbound sFlow:

- "Configuration considerations for ACL-based Inbound sFlow"
- "Creating an ACL with an sFlow clause"
- "Specifying an sFlow collector"

Configuration considerations for ACL-based Inbound sFlow

The followings sections describe the configuration considerations for ACL-based Inbound sFlow:

- sFlow must be enabled on the router.
- **ACL-based mirroring:** The **mirror** and **copy-sflow** keywords are mutually exclusive on a per ACL clause basis.
- **Port-based monitoring:** Port-based monitoring and ACL-based sFlow can co-exist on the same interface.
- **Port-based sFlow:** Port and ACL-based sFlow can co-exist on the same interface. When both features are configured on an interface, packets that qualify as ACL-based sFlow packets are sent to the collector as ACL sample packets. Also, the user can configure ACL-based sFlow on an interface without configuring port-based sFlow.

- **IP Receive ACLs:** IP Receive ACLs are used for filtering or rate-limiting management traffic. The keyword **copy-sflow** is also supported for IP Receive ACLs.
- **Policy Based Routing:** The **copy-sflow** keyword is applicable for PBR ACLs.
- **IPv4 ACL based Rate-Limiting:** When the **copy-sflow** keyword is used in an IPv4 Rate Limiting ACL, only traffic permitted by the Rate Limiting engine is copied to the CPU for forwarding to the sFlow collector.
- **IPv4 ACLs on VRF endpoints:** You can apply ACL-based sFlow for VRF endpoints however such packets are treated as regular sampled sFlow packets and do not carry proprietary encapsulation. This can create a minor skew of statistics projection.
- **L2 ACLs:** The **copy-sflow** keyword is not supported for L2 ACLs.
- If the **copy-sflow** keyword is used for a clause that is applied to the outbound direction, it is ignored.

Creating an ACL with an sFlow clause

The **copy-sflow** keyword has been added for inclusion in IPv4 and IPv6 ACL clauses to direct traffic that meets the criteria in the clause to be sent to the sFlow collector. In the following example, the ACL is used to direct syn-ack packets sent from a server at address 10.10.10.1.

```
access-list 151 permit tcp host 10.10.10.1 any established syn copy-sflow
access-list 151 permit any any
```

The **copy-sflow** parameter directs selected traffic to the sFlow collector. Traffic can only be selected using the **permit** clause.

You must apply the ACL to an interface using the **ip access-group command** as shown in the following.

```
NetIron(config)# int eth 1/1
NetIron(config-if-e10000-1/1)# ip access-group 151 in
```

Specifying an sFlow collector

sFlow exports traffic statistics to an external collector. You can specify up to four collectors. You can specify more than one collector with the same IP address if the UDP port numbers are unique. You can have up to four unique combinations of IP address and UDP port number.

NOTE
sflow is implemented in the default VRF only. Therefore, sflow data is only accessible by the sflow collector (sflow destination host(s)) defined in the default VRF.

To specify sFlow collectors, enter a command such as the following.

```
NetIron(config)# sflow destination 10.10.10.1
```

This command specifies a collector with IP address 10.10.10.1, listening for sFlow data on UDP port 6343.

Syntax: [no] sflow destination <*ip-addr*> [<*dest-udp-port*>]

The <*ip-addr*> parameter specifies the collector's IP address.

The <*dest-udp-port*> parameter specifies the UDP port on which the sFlow collector will be listening for exported sFlow data. The default port number is 6343.

Only inbound traffic is selected using sFlow. This applies to both standard sFlow and ACL-based sFlow.

NOTE
The sampling rate is the average ratio of the number of packets incoming on an sFlow-enabled port, to the number of flow samples taken from those packets. However for ACL based sFlow, every matching packet is sent to the CPU. Consequently, configured sampling rates do not affect ACL based sFlow.

Displaying sFlow information

To display sFlow configuration information and statistics, enter the following command at any level of the CLI.

```
NetIron(config)# show sflow
sFlow services are enabled.
sFlow agent IP address: 30.30.30.2
Collector IP 10.10.10.1, UDP 6343
Polling interval is 20 seconds.
Configured default sampling rate: 1 per 2048 packets.
0 UDP packets exported
0 sFlow samples collected.
sFlow ports   Global Sample Rate   Port Sample Rate   Hardware Sample Rate
        3/1                2048               2048   2048
        3/2                2048               2048   2048
        3/3                2048               2048   2048
        3/4                2048               2048   2048
```

Syntax: show sflow

This command shows the following information.

TABLE 409 sFlow information

This field...	Displays...
sFlow services	The feature state, which can be one of the following: • disabled • enabled
sFlow agent IP address	The IP address that sFlow is using in the agent_address field of packets sent to the collectors. Refer to "Source address" on page 2130.
Collector	The collector information. The following information is displayed for each collector: • IP address • UDP port If more than one collector is configured, the line above the collectors indicates how many have been configured.
Polling interval	The port counter polling interval.
Configured default sampling rate	The configured global sampling rate. If you changed the global sampling rate, the value you entered is shown here. The actual rate calculated by the software based on the value you entered is listed on the next line, "Actual default sampling rate".
UDP packets exported	The number of sFlow export packets the PowerConnect has sent. **NOTE:** Each UDP packet can contain multiple samples.

TABLE 409 sFlow information (Continued)

This field...	Displays...
sFlow samples collected	The number of sampled packets that have been sent to the collectors.
sFlow ports	The ports on which you enabled sFlow.
Global Sample Rate	The global sampling rate for the PowerConnect.
Port Sampling Rates	The sampling rates of a port on which sFlow is enabled.
Hardware Sample Rate	The actual sampling rate. This is the same as the Global Sample Rate

Displaying ACL-based sFlow statistics

Use the show sflow command to display the number of sFlow samples collected for ACL-based sFlow. These statistics are shown in bold in the following display.

```
NetIron# show sflow
sFlow services are disabled.
sFlow agent IP address: 10.10.10.254
Collector IP 10.10.10.1, UDP 6343
Polling interval is 30 seconds.
Configured default sampling rate: 1 per 1024 packets.
0 UDP packets exported
0 sFlow samples collected.
5 ACL sFlow samples collected
sFlow ports   Global Sample Rate   Port Sample Rate   Hardware Sample Rate
        4/1                1024               8192   8192
```

All other display parameters are unchanged as documented in the PowerConnect User Guide.

Clearing sFlow statistics

To clear the UDP packet and sFlow sample counters in the **show sflow** display, enter the following command.

```
NetIron(config)# clear statistics
```

Syntax: clear statistics

This command clears the values in the following fields of the **show sflow** display:

- UDP packets exported
- sFlow samples collected

NOTE
This command also clears the statistics counters used by other features.

Chapter

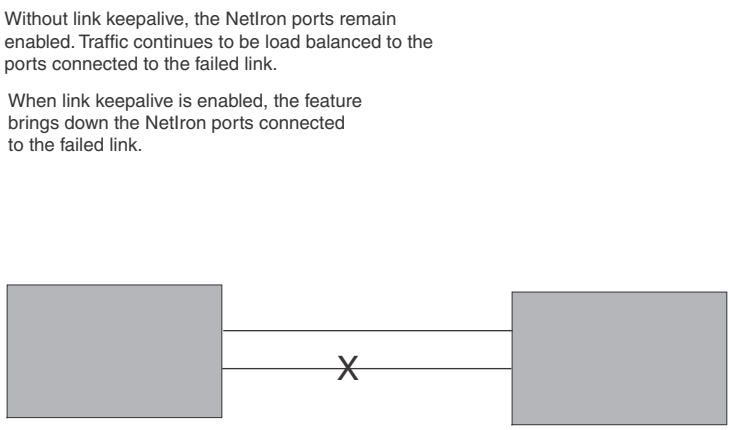
Configuring Uni-Directional Link Detection (UDLD) 58

The following Uni-Directional Link Detection (UDLD) features are supported by PowerConnect B-MLXe Series.

- Uni-Directional Link Detection (UDLD)
- UDLD for Tagged Ports

Uni-directional Link Detection (UDLD) monitors a link between two PowerConnect devices and provides a fast detection of link failures. UDLD brings the ports on both ends of the link down if the link goes down at any point between the two devices. This feature is useful for links that are individual ports and for LAG links. Figure 233 shows an example.

FIGURE 233 UDLD example



By default, ports enabled for UDLD exchange proprietary health-check packets once every 500 ms (the keepalive interval). If a port does not receive a health-check packet from the port at the other end of the link within the keepalive interval, the port waits for four more intervals. If the port still does not receive a health-check packet after waiting for five intervals, the port concludes that the link has failed and takes the port down.

The Keepalive Interval and Keepalive Retries can be configured to values other than the default, as shown in "Changing the keepalive interval" on page 2142 and "Changing the keepalive retries" on page 2142.

Configuration considerations

This section describes the configuration considerations:

- The feature is supported only on Ethernet ports.
- To configure UDLD on a LAG group, you must configure the feature on each port of the group individually. Configuring UDLD on a LAG group's primary port enables the feature on that port only.

- Dynamic LAG is not supported. If you want to configure a LAG group that contains ports on which UDLD is enabled, you must remove the UDLD configuration from the ports. After you create the LAG group, you can re-add the UDLD configuration.

Configuring UDLD

To enable UDLD on a port, enter a command such as the following at the global CONFIG level of the CLI.

```
NetIron(config)# link-keepalive ethernet 1/1
```

Syntax: [no] link-keepalive ethernet <*slot*>/<*portnum*> **[ethernet** <*slot*>/<*portnum*>**]**

To enable the feature on a LAG group, enter commands such as the following.

```
NetIron(config)# link-keepalive ethernet 1/1 ethernet 1/2
NetIron(config)# link-keepalive ethernet 1/3 ethernet 1/4
```

These commands enable UDLD on ports 1/1 – 1/4. You can specify up to two ports on the same command line.

Changing the keepalive interval

By default, ports enabled for UDLD send a link health-check packet once every 500 ms. You can change the interval to a value from 1 – 60, where 1 is 100 ms, 2 is 200 ms, and so on. To change the interval, enter a command such as the following.

```
NetIron(config)# link-keepalive interval 3
```

Syntax: [no] link-keepalive interval <*num*>

The <*num*> parameter specifies how often the ports send a UDLD packet. You can specify from 1 – 60, in 100 ms increments. The default is 5 (500 ms).

Changing the keepalive retries

You can change the maximum number of keepalive attempts to a value from 3 – 10. To change the maximum number of attempts, enter a command such as the following.

```
NetIron(config)# link-keepalive retries 4
```

Syntax: [no] link-keepalive retries <*num*>

The <*num*> parameter specifies the maximum number of times the port will try the health check. You can specify a value from 3 – 10. The default is 5.

UDLD for tagged ports

The default implementation of UDLD sends the packets untagged, even across tagged ports. If the untagged UDLD packet is received by a third-party switch, that switch may reject the packet. As a result, UDLD may be limited only to Dell devices, since UDLD may not function on third-party switches.

Beginning with Multi-Service IronWare release 04.1.00, you can configure ports to send out UDLD control packets that are tagged with a specific VLAN ID as tagged UDLD control packets. The enhancement also allows third party switches to receive the control packets that are tagged with the specified VLAN.

To allow ports to receive and send UDLD control packets tagged with a specific VLAN ID, enter commands such as the following:

```
NetIron(config)# link-keepalive ethernet 1/18 vlan 22
```

This commands enables UDLD on port 1/18 and allows UDLD control packet tagged with VLAN 22 to be received and sent on port 1/18.

Syntax: [no] link-keepalive ethernet <portnum> [vlan <vlan-ID>]

Enter the slot number (if applicable) and the port number of the Ethernet port.Enter the ID of the VLAN that the UDLD control packets can contain to be received and sent on the port. If a VLAN ID is not specified, then UDLD control packets are sent out of the port as untagged packets.

NOTE
You must configure the same VLANs that will be used for UDLD on all devices across the network; otherwise, the UDLD link cannot be maintained

Displaying UDLD information

Displaying information for all ports

.In the following example, port 1/1 is tagged in VLAN 2 and is configured for tagged UDLD. Port 1/2 is not configured for tagged UDLD:

```
PowerConnect#show link-keepalive
Total link-keepalive enabled ports: 2
Keepalive Retries: 5    Keepalive Interval: 5 * 100 MilliSec.

Port    Physical Link   Link-keepalive  Logical link    Link-vlan
1/1         down            down                down              2
1/2         down            down                down
PowerConnect#
```

Syntax: show link-keepalive [ethernet <*slot*>/<*portnum*>**]**

TABLE 410 CLI display of UDLD information

This field...	Displays...
Total link-keepalive enabled ports	The total number of ports on which UDLD is enabled.
Keepalive Retries	The number of times a port will attempt the health check before concluding that the link is down.
Keepalive Interval	The number of seconds between health check packets.
Port	The port number.
Physical Link	The state of the physical link. This is the link between the PowerConnect port and the directly connected device.
Link-keepalive	Show if the keepalive link is up or down.

TABLE 410 CLI display of UDLD information (Continued)

This field...	Displays...
Logical Link	The state of the logical link. This is the state of the link between this PowerConnect port and the PowerConnect port on the other end of the link. If the states of both Physical Link and Link-keepalive are up, then Logical link is up. If either or both Physical Link and Link-keepalive states are down, then Logical Link displays "down".
Link-vlan	The VLAN that the port is configured to tag the UDLD packets with.

If a port is disabled by UDLD, the change also is indicated in the output of the **show interfaces brief** command. Here is an example.

```
NetIron(config)# show interface brief

Port  Link State      Dupl Speed Trunk Tag Priori MAC            Name
1/1   Up   LK-DISABLE None None  None  No  level0 00e0.52a9.bb00
1/2   Down None       None None  None  No  level0 00e0.52a9.bb01
1/3   Down None       None None  None  No  level0 00e0.52a9.bb02
1/4   Down None       None None  None  No  level0 00e0.52a9.bb03
```

If the port was already down before you enabled UDLD for the port, the port's state is listed as None.

Syntax: show interface brief

The **show link-keepalive** command shows the following.

```
NetIron(config)# show link-keepalive ethernet
Current State   : down          Remote MAC Addr  : 0000.0000.0000
Local Port      : 1/1           Remote Port      : n/a
Local System ID : e0eb8e00      Remote System ID : 00000000
Packets sent    : 0             Packets received : 0
Transitions     : 0
```

Syntax: show link-keepalive ethernet

Displaying information for a single port

To display detailed UDLD information for a specific port, the command **show link-keepalive ethernet 4/1** is valid for a port that is not configured for Tagged UDLD. For a Tagged UDLD port VLAN information is also included. In the following example, port 1/1 is tagged in VLAN 2 and is configured for tagged UDLD. Port 1/2 is not configured for tagged UDLD.

```
PowerConnect#show link-keepalive ethernet 1/1

Current State   : down          Remote MAC Addr  : 0000.0000.0000
Local Port      : 1/1           Remote Port      : n/a
Local System ID : 1bb3d340      Remote System ID : 00000000
Packets sent    : 0             Packets received : 0
Transitions     : 5             Link-Vlan        :2

PowerConnect#show link-keepalive ethernet 1/2

Current State   : down          Remote MAC Addr  : 0000.0000.0000
Local Port      : 1/2           Remote Port      : n/a
```

```
Local System ID : 1bb3d340      Remote System ID : 00000000
Packets sent    : 0             Packets received : 0
Transitions     : 6

PowerConnect#
```

TABLE 411 CLI display of detailed UDLD information

This field...	Displays...
Current State	The state of the logical link. This is the link between this PowerConnect port and the PowerConnect port on the other end of the link.
Remote MAC Addr	The MAC address of the port or device at the remote end of the logical link.
Local Port	The port number on this PowerConnect router.
Remote Port	The port number on the PowerConnect router at the remote end of the link. **NOTE:** The Remote Port number shown in this parameter reflects the port ID sent by the other router or switch and interpreted by this local router. In cases where this router interprets the port ID different than the router that sent the port ID, the port shown can be incorrect.
Local System ID	A unique value that identifies this PowerConnect router. The ID can be used by Dell technical support for troubleshooting.
Remote System ID	A unique value that identifies the PowerConnect router at the remote end of the link.
Packets sent	The number of UDLD health-check packets sent on this port.
Packets received	The number of UDLD health-check packets received on this port.
Transitions	The number of times the logical link state has changed between up and down.
Link-vlan	The VLAN that the port is configured to tag the UDLD packets with.

The **show interface ethernet** <*slot*>/<*portnum*> command also displays the UDLD state for an individual port. In addition, the line protocol state listed in the first line will say “down” if UDLD has brought the port down. Here is an example.

```
NetIron(config)# show interface ethernet 1/1
GigabitEthernet2/1 is disabled, line protocol is down, link keepalive is enabled
  Hardware is GigabitEthernet, address is 000c.dbe2.5900 (bia 000c.dbe2.5900)
  Configured speed 1Gbit, actual unknown, configured duplex fdx, actual unknown
  Configured mdi mode AUTO, actual unknown
  Member of 2 L2 VLANs, port is tagged, port state is Disabled
  STP configured to ON, Priority is level7, flow control enabled
  Force-DSCP disabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  MTU 1522 bytes, encapsulation ethernet
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 0 packets
  0 packets output, 0 bytes, 0 underruns
  Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions, DMA transmitted 0 packets
```

In this example, the port has been brought down by UDLD. Notice that in addition to the information in the first line, the port state on the fourth line of the display is listed as DISABLED.

Clearing UDLD statistics

To clear UDLD statistics, enter the following command.

```
NetIron# clear link-keepalive statistics
```

Syntax: clear link-keepalive statistics

This command clears the Packets sent, Packets received, and Transitions counters in the **show link keepalive ethernet** *<slot>/<portnum>* display.

Chapter

BiDirectional Forwarding Detection (BFD) 59

PowerConnect B-MLXe supports the following BiDirectional Forwarding Detection (BFD) features:

- BiDirectional Forwarding Detection (BFD)
- BFD for IPv4 IS-IS
- BFD for IPv6 IS-IS
- BFD for OSPFv2
- BFD for OSPFv3
- BFD for IPv4 BGP
- Configuring BFD for RSVP-TE LSPs
- IP router alert option
- MPLS BFD

Multi-Service IronWare software provides support for Bidirectional Forwarding Detection (BFD). BFD provides rapid detection of the failure of a forwarding path by checking that the next-hop device is alive. Without BFD enabled, it can take from 3 to 30 seconds to detect that a neighboring device is not operational causing packet loss due to incorrect routing information at a level unacceptable for real-time applications such as VOIP and video over IP. Using BFD, you can detect a forwarding path failure in 300 milliseconds or less depending on your configuration.

A BFD session is automatically established when a neighbor is discovered for a protocol provided that BFD is enabled on the interface on which the neighbor is discovered and BFD is also enabled for the protocol (by interface or globally). Once a session is set-up, each device transmits control messages at a high rate of speed that is negotiated by the devices during the session setup. To provide a detection time of 150 milliseconds, it is necessary to process 20 messages per second of about 70 to 100 bytes each per each session. A similar number of messages also need to be transmitted out per each session. Once a session is set-up, that same message is continuously transmitted at the negotiated rate and a check is made that the expected control message is received at the agreed frequency from the neighbor. If the agreed upon messages are not received from the neighbor within a short period of time, the neighbor is considered to be down.

NOTE
BFD session establishment on an interface does not start until 90 seconds after the interface comes up. The reason for this delay is to ensure that the link is not effected by unstable link conditions which could cause BFD to flap. This delay time is not user configurable.

The BFD Control Message is an UDP message with destination port 3784.

NOTE
BFD version 0 is not supported in this implementation and BFD version 1 is not compatible with BFD version 0.

NOTE
BFD supports multi-slot LAGs in cases where all BFD packet are transmitted only on a single path which does not change unless the LAG active membership changes. BFD is not be supported on multi-slot LAGs where per-packet switching is used such that the path taken by the BFD packets will vary per packet.

NOTE
When BFD is configured with stringent values of 100/300 msec, BFD may flap when learning a large number of routes.

Number of BFD sessions supported

The devices have a set limit of 250 BFD sessions per system with a maximum number of 40 sessions per Interface Module. This number is inclusive of the fact that IS-IS and OSPF sessions on an Interface Module will include both Tx and Rx sessions. Consequently, the 40 sessions per Interface Module actually corresponds to 80 sessions where each OSPF and IS-IS session consumes 2 sessions (1 Tx and 1 Rx).

Unlike IS-IS and OSPF however, the Tx and Rx sessions for MPLS BFD can reside on different interface modules. This means that when counting MPLS BFD sessions against the Interface Module maximum, the Tx and Rx sessions must be counted separately. In practice this means that the maximum number of sessions per-Interface Module is 80; where each Tx and Rx session for MPLS BFD is counted as 1 and IS-IS and OSPF BFD sessions are counted as 2 towards a per-Interface Module maximum number of sessions of 80.

Configuring BFD parameters

When you configure BFD you must set timing and interval parameters. These are configured on each interface. When two adjacent interfaces with BFD are configured, they negotiate the conditions for determining if the connection between them is still active. The following command is used to set the BFD parameters:

```
NetIron(config-if-e1000-3/1)# bfd interval 100 min-rx 100 multiplier 3
```

Syntax: [**no**] **bfd interval** <*transmit-time*> **min-rx** <*receive-time*> **multiplier** <*number*>

The <*transmit-time*> variable is the interval in milliseconds between which this device will send a BFD message to its peer informing it that it is still operational. This value is specified in milliseconds. Acceptable values are: 50 - 30000.

The <*receive-time*> variable is the interval in milliseconds that this device waits to receive a BFD message from its peer. The device waits for this interval for the number of times specified in the <*number*> variable before determining that the connection to its peer in not operational. Acceptable values are: 50 - 30000.

NOTE
The **transit-time** and **receive-time** variables are the intervals desired by the local device. The actual values in use will be the negotiated values.

The <*number*> variable specifies the number of times in a single sequence that this device will wait to receive a BFD message from its peer before determining that the connection to that peer is not operational. Acceptable values are: 3 - 50.

Disabling BFD Syslog messages

Syslog messages are generated for BFD operations. These messages are described in "Syslog messages BFD". Logging of these messages is enabled by default. To disable logging of BFD messages use the following command:

```
NetIron(config)# no logging enable bfd
```

Syntax: [no] **logging enable bfd**

BFD logging is enabled by default. If you disable BFD logging as shown, you can re-enable it by using the **logging enable bfd** command.

Displaying BFD information

You can display BFD information for the device you are logged-in to and for BFD-configured neighbors as described in the following sections.

Displaying BFD information

The following example illustrates the output from the **show bfd** command:

```
NetIron# show bfd
 BFD State: ENABLED Version: 1
Current Registered Protocols: ospf/0  ospf6/0
All Sessions: Current: 4 Maximum Allowed: 100 Maximum Exceeded Count: 0
LP Sessions: Maximum Allowed on LP: 40  Maximum Exceeded Count for LPs: 0
LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions
1   4/4           2   2/2           3  0/0           4  0/0
5   0/0           6   0/0           7  0/0           8  0/0
9   0/0           10  0/0           11 0/0           12 0/0
13  0/0           14  0/0           15 0/0           16 0/0
BFD Enabled ports count: 2
Port       MinTx       MinRx       Mult Sessions
eth 2/1    100         100         3    2
pos 3/1    100         100         3    2
```

Syntax: show bfd

This display shows the following information.

TABLE 412 Display of BFD information

This field...	Displays...
BFD State	Specifies if BFD is Enabled or Disabled on the device.
Version	Specifies the version of the BFD protocol operating on the device.
Current Registered Protocols	Specifies which protocols are registered to use BFD on the device. Possible values are mpls/0, ospf/0, ospf6/0, or isis_task/0
All Sessions	

TABLE 412 Display of BFD information (Continued)

This field...	Displays...
Current:	The number of BFD sessions currently operating on the device.
Maximum Allowed	The maximum number of BFD sessions that are allowed on the device.
Maximum Exceeded Count	The number of times the request to set up a BFD session was declined because it would have resulted in exceeding the maximum number of BFD sessions allowed on the device.
LP Sessions:	
Maximum Allowed on LP	The maximum number of BFD sessions that are allowed on an interface module.
Maximum Exceeded Count for LPs	The number of times the request to set up a BFD session was declined because it would have resulted in exceeding the maximum number of BFD sessions allowed on an interface module.
LP	The number of the interface module that the Current Session Count is displayed for.
Sessions	The number of Transmit (Tx) and Receive (Rx) BFD sessions currently operating on the specified interface module.
BFD Enabled ports count	The number of ports on the device that have been enabled for BFD.
Port	The port that BFD is enabled on.
MinTx	The interval in milliseconds between which the device desires to send a BFD message from this port to its peer.
MinRx	The interval in milliseconds that this device desires to receive a BFD message from its peer on this port.
Mult	The number of times that the device will wait for the MinRx time on this port before it determines that its peer device is non-operational.
Sessions	The number of BFD sessions originating on this port.

Displaying BFD neighbor information

The following example illustrates the output from the **show bfd neighbor** command.

```
NetIron# show bfd neighbor
Total number of Neighbor entries: 2
NeighborAddress                         State   Interface Holddown  Interval  RH
12.14.1.1                               UP      eth 3/1   300000    100000    1
12.2.1.1                                UP      eth 2/1   300000    100000    1
```

Syntax: **show bfd neighbor** [**interface ethernet** <*slot/port*> | **interface pos** <*slot/port*> | **interface ve** <*port-no*>]

The **interface ethernet** option displays BFD neighbor information for the specified ethernet interface only.

The **interface ve** option displays BFD neighbor information for the specified virtual interface only.

This display shows the following information.

TABLE 413 Display of BFD information

This field...	Displays...
Total number of Neighbor entries	The number of neighbors that have established BFD sessions with ports on this device.
NeighborAddress	The IPv4 or IPv6 address of the remote peer.
State	The current state of the BFD session: Up Down A.DOWN – The administrative down state. INIT – The Init state. UNKNOWN – The current state is unknown.
Interface	The logical port (physical or virtual port) on which the peer is known. The physical port can be either Ethernet or POS.
Holddown	The interval in microseconds after which the session will transition to the down state if no message is received.
Interval	The interval in microseconds at which the local device sends BFD messages to the remote peer.
RH	Heard from remote.

To display BFD Neighbor information in the detailed format use the following command.

```
NetIron# show bfd neighbor details
Total number of Neighbor entries: 1
NeighborAddress                         State   Interface Holddown  Interval  RH
12.14.1.1                               UP      ve 50     300000    100000    1
  Registered Protocols(Protocol/VRFID): ospf/0
  Local: Disc: 1, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 100000, MinRxInterval: 100000, Multiplier: 3
  Remote: Disc: 22, Diag: 7, Demand: 0 Poll: 0
          MinTxInterval: 100000, MinRxInterval: 100000, Multiplier: 3
  Stats: RX: 72089 TX: 72101 SessionUpCount: 1 at SysUpTime: 0:1:30:54.775
  Session Uptime: 0:1:30:6.375, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port: eth 4/1, Vlan Id: 50
```

Syntax: show bfd neighbor details [<*ip-address*> | <*ipv6-address*>]

This display shows the following information.

TABLE 414 Display of BFD neighbor detail information

This field...	Displays...
Total number of Neighbor entries	Total number of BFD sessions.
NeighborAddress	IPv4 or IPv6 address of the remote peer.
State	The current state of the BFD session: Up Down A.DOWN – The administrative down state. INIT – The Init state. UNKNOWN – The current state is unknown.
Interface	The logical port on which the peer is known.

TABLE 414 Display of BFD neighbor detail information (Continued)

This field...	Displays...
Holddown	The interval in microseconds after which the session will transition to the down state if no message is received.
Interval	The interval in microseconds at which the local device sends BFD messages to the remote peer.
RH	Heard from remote.
Registered Protocols	Specifies which protocols are registered to use BFD on this port.
Local:	
Disc	Value of the “local discriminator” field in the BFD Control Message as used by the local device in the last message sent.
Diag	Value of the “diagnostic” field in the BFD Control Message as used by the local device in the last message sent.
Demand	Value of the “demand” bit in the BFD Control Message as used by the local device in the last message sent.
Poll	Value of the “poll” bit in the BFD Control Message as used by the local device in the last message sent.
MinTxInterval	The interval in microseconds between which the device will send a BFD message from this local neighbor port to its peer.
MinRxInterval	The interval in microseconds that the neighbor device waits to receive a BFD message from its peer on this local port.
Multiplier	The number of times that the neighbor device will wait for the MinRxInterval time on this port before it determines that its peer device is non-operational.
Remote:	
Disc	Value of the “local discriminator” field in the BFD Control Message as received in the last message sent by the remote peer.
Diag	Value of the “diagnostic” field in the BFD Control Message as received in the last message sent by the remote peer.
Demand	Value of the “demand” bit in the BFD Control Message as received in the last message sent by the remote peer.
Poll	Value of the “poll” bit in the BFD Control Message as received in the last message sent by the remote peer.
MinTxInterval	The interval in milliseconds between which the device will send a BFD message from the remote neighbor port to its peer.
MinRxInterval	The interval in milliseconds that the neighbor device waits to receive a BFD message from its peer on this remote port.
Multiplier	The number of times that the remote neighbor device will wait for the MinRxInterval time on this port before it determines that its peer device is non-operational.
Stats: Rx	Total number of BFD control messages received from the remote peer.
Stats: Tx	Total number of BFD control messages sent to the remote peer.
Stats: SessionUpCount	The number of times the session has transitioned to the UP state.
Stats: SysUpTime	The amount of time that the system has been up.

TABLE 414 Display of BFD neighbor detail information (Continued)

This field...	Displays...
Session Uptime	The amount of time the session has been in the UP state.
LastSessionDownTimestamp	The system time at which the session last transitioned from the UP state to some other state.
Physical Port	The physical port on which the peer is known.
Vlan Id	The VLAN ID of the VLAN that the physical port is resident on.

Clearing BFD neighbor sessions

You can clear all BFD neighbor sessions or a specified BFD neighbor session using the following command.

```
NetIron# clear bfd neighbor
```

Syntax: clear bfd neighbor [<*ip-address*> | <*ipv6-address*>]

The <*ip-address*> variable specifies the IPv4 address of a particular neighbor whose session you want to clear BFD.

The <*ipv6-address*> variable specifies the IPv6 address of a particular neighbor whose session you want to clear BFD.

Executing this command without specifying an IP or IPv6 address clears the sessions of all BFD neighbors.

Configuring BFD for the specified protocol

BFD can be configured for use with the following protocols:

- OSPFv2
- OSPFv3
- IS-IS
- BGP

NOTE
BFD is not supported for OSPF v2 or v3 virtual links.

NOTE
BFD brings ISIS and OSPF down with it when RSTP path-cost changes are made to the switch Alt Discarding port.

Configuring BFD for OSPFv2

You can configure your device for BFD on the OSPFv2 protocol for all OSPFv2 enabled interfaces or for specific interfaces as shown in the following sections.

Enabling BFD for OSPFv2 for all interfaces

You can configure BFD for OSPFv2 on all of a device's OSPFv2 enabled interfaces using the command shown in the following"

```
NetIron# device ospf
NetIron(config-ospf-device)# bfd all-interfaces
```

Syntax: [**no**] **bfd all-interfaces**

Although this command configures BFD for OSPFv2 on all of the OSPFv2 enabled interfaces for a device, it is not required if you use the **ip ospf bfd** command to configure specific interfaces. It can be used independently or together with the **ip ospf bfd** command.

Enabling or disabling BFD for OSPFv2 for a specific interface

You can selectively enable or disable BFD on any OSPFv2 interface as shown.

```
NetIron# (config-if-e1000-3/1)# ip ospf bfd disable
```

Syntax: ip ospf bfd [**disable** | **enable**]

The **disable** option disables BFD for OSPFv3 on the interface. The enable option enables BFD for OSPFv3 on the interface

Configuring BFD for OSPFv3

You can configure your device for BFD on the OSPFv3 protocol for all OSPFv3 enabled interfaces or for specific interfaces as shown in the following sections.

Enabling BFD for OSPFv3 for all interfaces

You can configure BFD for OSPFv3 on all OSPFv3 enabled interfaces using the command shown in the following.

```
NetIron(config)# ipv6 router ospf
NetIron(config-ospf6-router)# bfd all-interfaces
```

Syntax: [**no**] **bfd all-interfaces**

Although this command configures BFD for OSPFv3 on all of the OSPFv3 enabled interfaces on a device, it is not required if you use the **ipv6 ospf bfd** command to configure specific interfaces. It can be used independently or together with the **ipv6 ospf bfd** command.

Enabling or disabling BFD for OSPFv3 for a specific interface

You can selectively enable or disable BFD on any OSPFv3 interface as shown in the following.

```
NetIron#(config-if-e1000-3/1)# ipv6 ospf bfd enable
```

Syntax: ipv6 ospf bfd [**disable** | **enable**]

The **disable** option disables BFD for OSPFv3 on the interface. The **enable** option enables BFD for OSPFv2 on the interface.

Configuring BFD for IS-IS

You can configure your device for BFD (for both IPv4 ad IPv6 IS-IS neighbors) for the IS-IS protocol for all IS-IS enabled interfaces, or for specific interfaces as shown in the following sections.

Enabling BFD for IS-IS for all interfaces

You can configure IS-IS for IS-IS on all S-IS enabled interfaces for a device using this command.

```
NetIron# router isis
NetIron(config-isis-router)# bfd all-interfaces
```

Syntax: [**no**] **bfd all-interfaces**

Although this command configures BFD for IS-IS on all IS-IS enabled interfaces on the device, it is not required if you use the **isis bfd** command to configure specific interfaces. It can be used independently or together with the **isis bfd** command.

Enabling or disabling BFD for IS-IS for a specific interface

You can selectively enable or disable BFD on any IS-IS interface as shown in the following.

```
NetIron#(config-if-e1000-3/1)# isis bfd enable
```

Syntax: [**no**] **isis bfd** [**enable** | **disable**]

The **enable** option enables IS-IS on the interface. The **disable** option disables BFD for IS-IS on the interface.

Configuring BFD for BGP4

You can configure your device for BFD for BGP4. BGP4 supports IPv4 and IPv6 IBGP and EBGP peers. These peers can be directly connected or multihop. BFD for BGP4 supports single hop and multihop BFD on Ethernet, POS and Virtual Interfaces. BFD for BGP4 is not supported on loopback, tunnel (including IGP shortcut) and management interfaces.

BFD for BGP4 global configuration – Using this configuration, you can enable and disable BFD BGP4 on a global level. In addition, you can use the global command to set revised default values for the transmit interval, receive interval, and for the detection time multiplier for all BFD multihop BGP4 sessions.

BFD for BGP4 at global, peer, and peer group configuration – Using this configuration, you can enable and disable BFD for individual peers or peer groups. You can also change the values for the transmit interval, receive interval, and for the detection time multiplier. If these values are not specified at this level, they are obtained from the values configured at the global level.

BFD for BGP4, which is disabled by default, can be enabled or disabled at the global BGP router level or for each individual peer or peer group. The hierarchy for BFD for BGP4 is as follows:

- Peer and peer group parameters can be configured but will not take effect until BFD for BGP4 has been enabled.
- Peer configurations will override global and peer group configurations.
- Peer group configurations will override global configurations
- The **bfd-enable** command under **router bgp** overrides all other BGP4 BFD configurations

Enabling BFD for BGP4 globally

By default, BFD for BGP4 is disabled and can be first enabled globally and then on each peer. To enable BFD for BGP4 globally, enter commands such as the following.

```
NetIron# router bgp
NetIron(config-bgp)# bfd-enable
```

To disable BFD for BGP globally and terminate all BFD sessions used by BGP4, enter commands such as the following

```
NetIron# router bgp
NetIron(config-bgp)# no bfd-enable
```

Syntax: [**no**] **bfd-enable**

NOTE
If BFD for BGP4 is globally disabled and then enabled, the original BFD sessions for BGP4 may not be available, depending on whether or not the maximum BFD sessions limit has been reached. When a BFD session for BGP4 is disabled, the session will be removed but BGP4 peering will not go down. The remote BFD peer will be informed that BFD use is disabled.

Setting the transmit, receive, and detection time multiplier at the global level

When using BFD for BGP4, you must configure BFD globally at the **router BGP** level. You can also use this configuration to set new default values for the transmit interval, receive interval, and for the detection time multiplier.

For a single hop EBGP session, the BFD parameters configured under interface will be used because the BFD session for single hop is also shared with other applications. To create a BFD session for a single hop BGP4 session, you must have BFD enabled and the timers configured for the interface on which single hop BGP4 peering is established.

NOTE
For multihop BFD sessions, BFD does not have to be enabled for any of the interfaces, and the BFD timers need not be configured, since the default values can be used.

The timers parameters **min-tx**, **min-rx** and **multiplier** can also be configured for each peer and peer group and will override the global configuration.

To configure a multi hop EBGP or IBGP session, enter a command such as the following.

```
NetIron(config)# router bgp
NetIron(config-bgp)# bfd-enable
NetIron(config-bgp)# bfd min-tx 500 min-rx 500 multiplier 5
```

Syntax: [**no**] **bfd** [**min-tx** <*transmit-time*> **min-rx** <*receive-time*> **multiplier** <*number*>]

The <*transmit-time*> variable is the interval in milliseconds between which this device will send a BFD message to its peer informing it that it is still operational. Acceptable values are: 50 - 30000. Default value: 1000 (unless changed at the global level

The <*receive-time*> variable is the interval in milliseconds that this device waits to receive a BFD message from its peer. The device will wait for this interval for the number of times specified in the <*number*> variable before determining that the connection to its peer in not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level)

The **multiplier** option allows you to specify a value for the number of times in a single sequence that this device will wait to receive a BFD message from its peer before determining that the connection to that peer is not operational. This value is set at the <*number*> variable. Acceptable values are 3 - 50. The default value is 3.

The **no** option globally removes the BFD for BGP4 configuration from the device.

Holdover interval

The BFD holdover interval is supported for both single hop and multihop sessions. It sets the time by which the BFD session DOWN notification to BGP4 is delayed. If within that holdover time, the BFD session is UP then BGP4 is not notified of the BFD session flap.

The holdover interval can be configured globally, on each peer, or peer-group.

If the **bfd holdover-interval** is set to 20 seconds, when a notification is received from BFD that the BFD session has moved to DOWN state, the system waits for 20 seconds before sending the BFD session down notification to BGP4 state machine. If the BFD session returns to UP state before the 20 seconds expires, the BGP4 state machine is not notified that the BFD session flapped. Otherwise, after 20 seconds the BFD session down notification is passed to the BGP4 state machine. If BFD for BGP4 is disabled, the request to not use BFD for BGP4 is passed to BFD by BGP4, BFD acknowledges this request, and the BFD session is removed. If BFD is disabled, BGP4 is notified and asks BFD to remove the single hop BFD session on the interface.

To configure the BFD down notification delay, enter a command such as the following.

```
NetIron(config-bgp)# bfd holdover-interval 20
```

Syntax: [**no**] **bfd holdover-interval** <*time-seconds*>

The <*time-seconds*> variable is a number between 0 and 30 seconds. The default is 0 seconds.

The **no** option removes the BFD for BGP4 holdover interval from the configuration.

Enabling BFD for a BGP4 peer group

To enable BFD and create a peer group for BGP4, you must first create the peer group, then enable BFD for the peer group by entering commands such as the following.

```
NetIron(config-bgp)# neighbor pg1 peer-group
NetIron(config-bgp)neighbor pg1 fail-over bfd-enable
```

Syntax: [**no**] **neighbor** <*name*> **peer group**

Syntax: [**no**] **neighbor** <*name*> **fail-over bfd-enable**

The <*name*> variable specifies peer-group name of a particular neighbor.
The **no** option removes the BFD for BGP4 peer group from the configuration.

Enabling BFD timers for a BGP4 peer group

To enable BFD timers for a BGP4 peer group, you must first create the peer group, then enable BFD timers for the peer group by entering commands such as the following.

```
NetIron(config-bgp)# neighbor pg1 peer-group
NetIron(config-bgp)neighbor pg1 bfd min-tx 500 min-rx 500 multiplier 5
```

Syntax: [**no**] **neighbor** <*name*> **peer group**

Syntax: [**no**] **neighbor** <*name*> **bfd min-tx** <*transmit-time* > **min-rx** <*receive-time*> **multiplier** <*number*>

The <*name*> variable specifies peer-group name of a particular neighbor.

The <*transmit-time*> variable is the interval in milliseconds during which this device will send a BFD message to its peer informing it that it is still operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The <*receive-time*> variable is the interval in milliseconds that this device waits to receive a BFD message from its peer. The device waits for this interval for the number of times specified in the <*number*> variable before determining that the connection to its peer in not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The **multiplier** option allows you to specify a value for the number of times in a single sequence that the device waits to receive a BFD message from its peer before determining that the connection to that peer is not operational. This value is set at the <*number*> variable. Acceptable values are 3 - 50. The default value is 3.

The **no** option removes the BFD timers for the peer group from the configuration.

Enabling BFD for a specific BGP4 peer

To enable BFD for BGP4 for a specific neighbor or peer, enter a command such as the following

```
NetIron(config-bgp)# neighbor 12.14.1.1 fail-over bfd-enable
```

Syntax: [**no**] **neighbor** <*ipv4-address* | *ipv6-address*> **fail-over bfd-enable**

The <*ipv4-address* | *pv6-address*> variables specify the IPv4 or IPv6 address of a particular neighbor or peer.

The **no** option removes the BFD for BGP4 peer from the configuration.

Disabling BFD for a specific BGP4 peer

To disable BFD for BGP4 for a specific peer, enter a command such as the following.

```
NetIron(config-bgp)# neighbor 12.14.1.1 fail-over bfd-disable
```

Syntax: [**no**] **neighbor** <*ipv4-address* I *ipv6-address*> **fail-over bfd-disable**

The <*ipv4-address* I *ipv6-address*> variables specify the IPv4 or IPv6 address of a particular neighbor or peer.

The **no** option removes the BFD specific peer from the configuration.

Enabling BFD timers for a specific BGP4 peer

To enable BFD timers for a specific neighbor or peer for BGP4, you must first configure the bfd timers, and set the holdover interval by entering commands such as the following.

```
NetIron(config-bgp)# neighbor 12.14.1.1 bfd min-tx 500 min-rx 500 multiplier 5
NetIron(config-bgp)# bfd holdover-interval 20
```

Syntax: [**no**] **neighbor** <*ipv4-address* I *ipv6-address*> **bfd min-tx** <*transmit-time* > **min-rx** <*receive-time*> **multiplier**

Syntax: [**no**] **bfd holdover-interval** <*time-seconds*>

The <*ipv4-address* I *ipv6-address*> variables specify the IPv4 or IPv6 address of a particular neighbor or peer.

The <*time-seconds*> variable is a number between 0 and 30 seconds. The default is 0 seconds.

The <*transmit-time*> variable is the interval in milliseconds between which this device will send a BFD message to its peer informing it that it is still operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The <*receive-time*> variable is the interval in milliseconds that this device waits to receive a BFD message from its peer. The device will wait for this interval for the number of times specified in the <*number*> variable before determining that the connection to its peer is not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The **multiplier** option allows you to specify a value for the number of times in a single sequence that this device will wait to receive a BFD message from its peer before determining that the connection to that peer is not operational. This value is set at the <*number*> variable. Acceptable values are 3 - 50. The default value is 3.

The **no** option removes the BFD for BGP configuration for the peer.

Displaying BFD for BGP4

You can display BFD for BGP4 information for the device you are logged in to and for BFD configured neighbors as described in the following sections.

Displaying BFD information

The following example illustrates the output from the **show bfd** command:

```
NetIron# show bfd
 BFD State: ENABLED Version: 1
 Current Registered Protocols: bgp/1 ospf6/0 ospf/0 bgp/0
 All Sessions: Current: 0 Maximum Allowed: 100 Maximum Exceeded Count: 0
 LP Sessions: Maximum Allowed on LP: 20 Maximum Exceeded Count for LPs: 0
   LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions
   1   0/0           2   0/0           3   0/0           4   0/0
   5   0/0           6   0/0           7   0/0           8   0/0
   9   0/0           10  0/0           11  0/0           12  0/0
   13  0/0           14  0/0           15  0/0           16  0/0
 BFD Enabled ports count: 0
```

Syntax: show bfd

This display shows the following information.

TABLE 415 Display of BFD information

This field...	Displays...
BFD State	Specifies if BFD is Enabled or Disabled on the device.
Version	Specifies the version of the BFD protocol operating on the device.
Current Registered Protocols	Specifies which protocols are registered to use BFD on the device. Possible values are mpls/0, ospf/0, ospf6/0, or isis_task/0 or bgp/0
All Sessions	
Current:	The number of BFD sessions currently operating on the device.
Maximum Allowed	The maximum number of BFD sessions that are allowed on the device.
Maximum Exceeded Count	The number of times the request to set up a BFD session was declined because it would have exceeded the maximum number of BFD sessions allowed on the device.
LP Sessions:	
Maximum Allowed on LP	The maximum number of BFD sessions allowed on an interface module.
Maximum Exceeded Count for LPs	The number of times the request to set up a BFD session was declined because it would have exceeded the maximum number of BFD sessions allowed on an interface module.
LP	The number of the interface modules for which the Current Session Count is displayed.
Sessions	The number of Transmit (Tx) and Receive (Rx) BFD sessions currently operating on the specified interface module.
BFD Enabled ports count	The number of ports that have been enabled for BFD.
Port	The port on which BFD is enabled.
MinTx	The interval in milliseconds during which the device sends a BFD message from this port to its peer.
MinRx	The interval in milliseconds during which this device can receive a BFD message from its peer on this port.

TABLE 415 Display of BFD information (Continued)

This field...	Displays...
Mult	The number of times the device will wait for the MinRx time on this port before it determines that the peer device is non-operational.
Sessions	The number of BFD sessions originating on this port.

Displaying BFD applications

The following example illustrates the output from the **show bfd applications** command.

```
NetIron# show bfd applications
  Registered Protocols Count: 4
     Protocol  VRFID     Parameter
     bgp       1          0
     ospf6     0          0
     ospf      0          0
     bgp       0          0
```

TABLE 416 Display of BFD applications information

This field...	Displays...
Protocol	Which protocols are registered to use BFD on the device.
VRFID	The VRFID of the protocol.
Parameter	The parameter value passed by the protocol during registration with BFD.

Displaying BFD for BGP neighbor information

The following example illustrates the output from the **show bfd neighbor bgp detail** command.

```
NetIron# show bfd neighbor bgp detail
Total Entries:4 R:RxRemote(Y:Yes/N:No)H:Hop(S:Single/M:Multi)
NeighborAddress                         State  Interface Holddown   Interval R/H
101.101.101.100                         UP     ve 3      3000000    1000000  Y/M
  Registered Protocols(Protocol/VRFID): bgp/0
  Local: Disc: 26, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Remote: Disc: 7, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 14682 TX: 12364 SessionUpCount: 1 at SysUpTime: 0:2:46:24.725
  Session Uptime: 0:1:37:50.600, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/1,RX: eth 1/1,Vlan Id: 3
NeighborAddress                         State  Interface Holddown   Interval R/H
100.100.100.100                         UP     ve 3      3000000    1000000  Y/M
  Registered Protocols(Protocol/VRFID): bgp/0
  Local: Disc: 27, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Remote: Disc: 8, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 14232 TX: 12046 SessionUpCount: 1 at SysUpTime: 0:2:46:24.725
  Session Uptime: 0:1:37:49.650, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/1,RX: eth 1/1,Vlan Id: 3
NeighborAddress                         State  Interface Holddown   Interval R/H
1.1.1.1                                 UP     ve 3      3000000    1000000  Y/M
  Registered Protocols(Protocol/VRFID): bgp/0
```

```
  Local: Disc: 28, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Remote: Disc: 9, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 15652 TX: 12044 SessionUpCount: 1 at SysUpTime: 0:2:46:24.725
  Session Uptime: 0:1:37:48.725, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/1,RX: eth 1/1,Vlan Id: 3
NeighborAddress                       State  Interface Holddown   Interval R/H
102.102.102.100                       UP     ve 3      3000000    1000000  Y/M
  Registered Protocols(Protocol/VRFID): bgp/0
  Local: Disc: 29, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Remote: Disc: 10, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 14232 TX: 12044 SessionUpCount: 1 at SysUpTime: 0:2:46:24.725
  Session Uptime: 0:1:37:48.550, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/1,RX: eth 1/1,Vlan Id: 3
NetIron#
```

Syntax: show bfd neighbor bgp [<*ipv4-address*> | <*ipv6-address*>| **detail**]

The <*ipv4-address*> | <*ipv6-address*> options display BFD neighbor information for the BGP specified IPv4 or IPv6 neighbor only.

The **detail** option displays BFD neighbor information for all BGP neighbors.

This display contains the following information.

TABLE 417 Display of BFD neighbor detail information

This field...	Displays...
Total number of Neighbor entries	Total number of BFD sessions.
NeighborAddress	IPv4 or IPv6 address of the remote peer.
State	The current state of the BFD session: Up Down A.DOWN – The administrative down state. INIT – The Init state. UNKNOWN – The current state is unknown.
Interface	The logical port on which the peer is known.
Holddown	The interval in microseconds after which the session will transition to the down state if no message is received.
Interval	The interval in microseconds at which the local device sends BFD messages to the remote peer.
RH	Heard from remote, values: Y or N where Y stand for Yes and N stand for No; H- Single hop/Multihop Values are S and M where S stand for Single Hop and M stand for MultiHop.
Registered Protocols	Specifies which protocols are registered to use BFD on this port.
Local:	
Disc	Value of the local discriminator field in the BFD Control Message as used by the local device in the last message sent.
Diag	Value of the diagnostic field in the BFD Control Message as used by the local device in the last message sent.

TABLE 417 Display of BFD neighbor detail information (Continued)

This field...	Displays...
Demand	Value of the demand bit in the BFD Control Message as used by the local device in the last message sent.
Poll	Value of the poll bit in the BFD Control Message as used by the local device in the last message sent.
MinTxInterval	The interval in microseconds during which the device will send a BFD message from this local neighbor port to the peer.
MinRxInterval	The interval in microseconds that the neighbor device waits to receive a BFD message from the peer on this local port.
Multiplier	The number of times the neighbor device will wait for the MinRxInterval time on this port before it determines the peer device is non-operational.
Remote:	
Disc	Value of the local discriminator field in the BFD Control Message as received in the last message sent by the remote peer.
Diag	Value of the diagnostic field in the BFD Control Message as received in the last message sent by the remote peer.
Demand	Value of the demand bit in the BFD Control Message as received in the last message sent by the remote peer.
Poll	Value of the poll bit in the BFD Control Message as received in the last message sent by the remote peer.
MinTxInterval	The interval in milliseconds during which the device will send a BFD message from the remote neighbor port to the peer.
MinRxInterval	The interval in milliseconds that the neighbor device waits to receive a BFD message from the peer on this remote port.
Multiplier	The number of times that the remote neighbor device will wait for the MinRxInterval time on this port before it determines that the peer device is non-operational.
Stats: Rx	Total number of BFD control messages received from the remote peer.
Stats: Tx	Total number of BFD control messages sent to the remote peer.
Stats: SessionUpCount	The number of times the session has transitioned to the UP state.
Stats: SysUpTime	The amount of time that the system has been up.
Session Uptime	The amount of time the session has been in the UP state.
LastSessionDownTimestamp	The system time at which the session last transitioned from the UP state to some other state.
Physical Port	The physical port on which the peer is known.
Vlan Id	The VLAN ID of the VLAN on which the physical port is resident.

.Displaying summary neighbor information

Support for BFD for BGP neighbors is highlighted in the bold text in the following output for the **show ip bgp neighbors** command.

```
Neighbor AS4 Capability Negotiation:
    As-path attribute count: 2
    Outbound Policy Group:
```

```
   ID: 1, Use Count: 3
BFD:Enabled,BFDSessionState:UP,Multihop:Yes
    LastBGP-BFDEvent:RX:Up,BGP-BFDError:No Error
    NegotiatedTime(msec):Tx:1000000,Rx:1000000,BFDHoldTime:3000000
    HoldOverTime(sec) Configured:22,Current:0,DownCount:0
TCP Connection state: ESTABLISHED, flags:00000044 (0,0)
 Maximum segment size: 1460
```

Configuring BFD for RSVP-TE LSPs

BFD can be configured for use with RSVP-LSPs to detect data plane failures for MPLS LSPs. Although the LSP ping facility can also be used for this purpose, BFD provides the following advantages:

- BFD provides faster failure detection because it does not required control plane verification, which is required by LSP ping.
- BFD can be used to detect faults on a large number of LSPs without requiring manual interaction, which is required by LSP ping.

BFD configuration for RSVP-TE LSPs is performed at the global and LSP levels, as described:

- **BFD for RSVP-TE LSPs global configuration** – allows you to enable and disable BFD on all of the RSVP-TE LSPs that have been configured for BFD at the LSP level. In addition, use the global command to set revised default values for the transmit interval, receive interval, and for the detection time multiplier for all BFD sessions on RSVP-TEs. This configuration command can also be used as a convenient method to turn BFD for MPLS on or off.
- **BFD for RSVP-TE LSPs configuration at the LSP level** – allows you to enable and disable BFD for individual RSVP-TE LSPs. You can also change the values for the transmit interval, receive interval, and for the detection time multiplier from the default values. If these values are not specified at this level, they are obtained from the values configured at the global level.

BFD, which is disabled by default, can be enabled or disabled at the global MPLS level, or for each individual LSP without affecting the LSP operational status. BFD parameters can also be changed without changing the state of the BFD session.

BFD for RSVP-TE LSPs operates with Fast ReRoute (FRR), Redundant, and Adaptive LSPs as described:

- **FRR LSPs –** Only one BFD session is created for an FRR LSP. A separate BFD session is not created for the detour path. When a switchover from a protected to a detour path occurs, the detour path resides on another interface module, and the BFD session is moved to that interface module. The BFD session can go down if the interface module has already reached the maximum number of BFD sessions. If the detour path is on the same interface module, the outgoing interface and label stack are updated on that interface module. A BFD session is not created for a detour path originated on a transit LSR.
- **Redundant LSPs –** One BFD session is created for the primary path of a redundant LSP. If the secondary path is in the hot-standby condition, a separate BFD session is created for it, but only if BFD is enabled on the secondary path. The two sessions operate independently.
- **Adaptive LSPs –** If a new instance of an adaptive LSP comes up on a different interface module, its BFD session is automatically created on that module. Otherwise, the local outgoing interface and label stack are updated on the existing interface module. When a BFD session is moved to a different interface module, the BFD session may be brought down if the interface module has already reached the maximum number of BFD sessions allowed on it.

BFD session support per-router and per-interface module

There is a limit to the number of BFD sessions available on a per-router and per-interface module basis as described:

- **per-router** – A maximum number of 250 BFD sessions are permitted per device
- **per-interface module** – A maximum number of 80 BFD sessions (Tx or Rx) are permitted per-interface module

These limitations are inclusive of any BFD sessions created for OSPFv2 or v3 and ISIS. If creating a BFD session will exceed these limits, the session will be denied. For a detailed description of how to calculate the number of BFD sessions supported, refer to "Number of BFD sessions supported" on page 2148.

BFD session creation

On ingress, one BFD session is created for each LSP after the LSP comes up. The BFD session status is then displayed in the output of the **show lsp** command. If the BFD session is not up, a failure reason is displayed in the output of the **show lsp** command. Possible reasons why a BFD session may fail to come up include exceeding the maximum supported number of BFD sessions, or if the global BFD configuration is disabled. If a BFD session does not come up because a BFD packet from the egress LSR is not received, an MPLS Echo Request with a BFD Discriminator TLV is resent until the session does come up. The retry timer is exponentially backed off.

On egress, a BFD session is created after the following sequence of events.

1. An MPLS Echo Request is received with a BFD Discriminator TLV
2. MPLS BFD is enabled globally
3. The maximum number of BFD sessions available on the device has not been reached.

NOTE
A BFD session created on an egress LSR is counted toward the maximum supported number of BFD sessions.

If the number of BFD sessions has reached the su pop rt ed maximum for the device, no MPLS Echo Reply or BFD control packet is sent. The ingress LSR will retry.

Because the source IP address cannot be changed for an MPLS BFD session after the session has come up, the LSR-ID is used as the source IP address for all MPLS BFD packets. This ensures that the session will not go down when an LSP path switch occurs.

BFD session down behavior

When a BFD session for an LSP goes down on an ingress LSR because the BFD detection time has expired, one of the following path switchovers will be triggered; from the protected path to the detour path, or from the primary path to the secondary path. In configurations with no alternative path, the LSP is brought down and the BFD session is deleted. The LSP then follows the normal retry procedures to come back up. On an egress LSR, a down BFD session does not have any impact on the RSVP session.

BFD session deletion

A BFD session is deleted when any of the following events occur:

- An LSP goes down
- BFD is disabled for the LSP
- BFD is disabled for all LSPs (using the global configuration)

These events are described in the following sections.

Enabling BFD for RSVP-TE LSPs at the global level

When using BFD for RSVP-TE LSPs, you must configure BFD globally at the **router mpls** level. You can also use this configuration to set new default values for the transmit interval, receive interval, and for the detection time multiplier, as shown.

```
NetIron(config)# router mpls
NetIron(config-mpls)# bfd
NetIron(config-mpls-bfd)# bfd min-tx 500 min-rx 500 multiplier 5
```

Syntax: [**no**] **bfd** [**min-tx** <*transmit-time*> **min-rx** <*receive-time*> **multiplier** <*number*>]

The <*transmit-time*> variable is the interval in milliseconds during which this device sends a BFD message to the peer informing it that it is still operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The <*receive-time*> variable is the interval in milliseconds the device waits to receive a BFD message from the peer. The device waits for the number of times specified in the <*number*> variable before determining that the connection to the peer in not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The **multiplier** option allows you to specify a value for the number of times in a single sequence that this device will wait to receive a BFD message from the peer before determining that the connection to that peer is not operational. This value is set at the <*number*> variable. Acceptable values are 3 - 50. The default value is 3.

The **no** option globally removes the BFD for RSVP-TE LSPs configuration from the device.

NOTE
BFD parameters configured globally can be changed dynamically without affecting the operational status of the LSP and the BFD session. When you make changes to the global configuration, the changes are applied to all egress MPLS BFD sessions. and only to the ingress BFD sessions with parameters that are derived from the global configuration.

Enabling BFD for a specific RSVP-TE LSPs

When you configure BFD globally, you must also configure it locally for the individual LSPs on which you want it to operate. You can also set separate values for the transmit interval, receive interval, and for the detection time multiplier for the specified LSP. The following example enables BFD for the LSP named blue and sets new parameter values.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp blue
NetIron(config-mpls-lsp-blue)# bfd
NetIron(config-mpls-lsp-blue-bfd)# bfd min-tx 500 min-rx 500 multiplier 5
```

Syntax: [**no**] **bfd min-tx** <*transmit-time*> **min-rx** <*receive-time*> **multiplier** <*number*>

The <*transmit-time*> variable is the interval in milliseconds during which this device sends a BFD message to the peer announcing that it is still operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The <*receive-time*> variable is the interval in milliseconds this device waits to receive a BFD message from the peer. The device waits for the number of times specified in the <*number*> variable before determining that the connection to the peer is not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The **multiplier** option allows you to specify a value for the number of times in a single sequence that this device waits to receive a BFD message from the peer before determining that the connection to that peer is not operational. This value is set at the <*number*> variable. Acceptable values are 3 - 50. The default value is 3.

The **no** option globally removes the BFD for RSVP-TE LSPs configuration from the device.

NOTE
BFD parameters configured for a specific LSP can be changed dynamically without affecting the operational status of the LSP and the BFD session.

Using this command you can also configure BFD for the secondary path of an LSP as shown in the following example.

```
NetIron(config)# router mpls
NetIron(config-mpls)# lsp blue
NetIron(config-mpls-lsp-blue)# secondary-path alt_sf_to_sj
NetIron(config-mpls-lsp-blue-sec-path)# bfd
```

Displaying MPLS BFD information

You can display the following information about an LSP BFD configuration:

- BFD Application Information
- BFD MPLS Information
- Detailed BFD MPLS Information
- MPLS BFD Global Configuration Information

You can also obtain MPLS BFD information using the **show bfd** command, as described in "Displaying BFD information" on page 2149, and the **show mpls lsp** command, as described in "Displaying signalled LSP status information" on page 1366.

Displaying BFD application information

The following example illustrates the output from the **show bfd application** command.

```
NetIron# show bfd application
 Registered Protocols Count: 3
     Protocol  VRFID     Parameter
     ospf       0           1
     ospf6      0           0
     isis_task  0           0
     mpls       0           0
```

This display shows the following information.

TABLE 418 Display of BFD application information

This field...	Displays...
Protocol	Specifies protocols registered to use BFD on the device. Possible values are mpls/0, ospf/0, ospf6/0, or isis_task/0
VRFID	The VRFID of the protocol.
Parameter	The parameter value passed by the protocol during registration with BFD.

Displaying BFD MPLS information

The following example shows output from the **show bfd mpls** command.

```
NetIron# show bfd mpls
Total number of MPLS BFD sessions: 2
Session name                          State   Interface Holddown  Interval  RH
lsp1                                  UP      eth 1/2   3000000   1000000   Y
11.11.11.1/1/22.22.22.2               UP      eth 1/2   3000000   1000000   Y
```

Syntax: show bfd mpls

This display shows the following information.

TABLE 419 Display of BFD MPLS information

This field...	Displays...
Total number of MPLS BFD Sessions	The number of BFD sessions that have been established on this device.
Session name	The name of the session: **For LSP Sessions –** the LSP name. **For RSVP Sessions –** the session-id which is displayed as IPv4 tunnel endpoint, tunnel ID, or extended tunnel ID.
State	The current state of the BFD session: Up Down A.DOWN – The administrative down state INIT – The Init state UNKNOWN – The current state is unknown
Interface	The logical port (physical or virtual port) on which the BFD packet is sent out. The physical port can be either an Ethernet, POS, or VE-enabled interface. The VE interface ID is specified by the <*vid*> variable.
Holddown	The interval in microseconds after which the session will transition to the down state if no message is received.
Interval	The interval in microseconds at which the local device sends BFD messages to the remote peer.
RH	Heard from remote.

Displaying BFD MPLS detailed information

The following example shows a display of BFD MPLS detailed information as a result of the **show bfd mpls detail** command. To view BFD MPLS information for a single LSP or RSVP session, use the **show bfd mpls lsp** command.

NOTE
The **show bfd mpls lsp** command displays the same information as the **show bfd mpls rsvp-session** command.

```
NetIron# show bfd mpls lsp lsp2
Session name                          State   Interface Holddown  Interval  RH
lsp2                                  UP      eth 1/2   3000000   1000000   Y
  Local: Disc: 3, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 500000, MinRxInterval: 500000, Multiplier: 3
  Remote: Disc: 3, Diag: 3, Demand: 1 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 305 TX: 305 SessionUpCount: 1 at SysUpTime: 0:0:4:46.200
  Session Uptime: 0:0:3:46.650, LastSessionDownTimestamp: 0:0:0:0.0
  Tx Port: eth 1/2, Rx Port: eth 1/2
```

Syntax: show bfd mpls [**detail** | **lsp** <*name*> | **rsvp-session** <*src-addr*> <*dest-addr*> <*tnnl-id*>]

This information shown in this display that is not defined in Table 419 is described in either Table 414 or Table 420.

TABLE 420 Display of BFD MPLS detail information

This field...	Displays...
Interface	The logical port (physical or virtual port) on which the BFD packet is sent out. The physical port can be either an Ethernet, POS, or VE-enabled interface. The VE interface ID is specified by the <*vid*> variable.
Tx Port:	The physical port on which the BFD packet is sent. When applicable, the Tx Port field displays a VE interface ID specified by the <*vid*> variable.
Rx Port:	The physical port on which the BFD packet is received.

Displaying MPLS BFD global configuration information

You can use the **show mpls bfd** command to display the global configuration information for a device, as shown in the following.

```
NetIron# show mpls bfd
MPLS BFD admin            = Enabled
Minimum TX interval       = 1000 msec
Minimum RX interval       = 1000 msec
Detection time multiplier = 3
```

Syntax: show mpls bfd

TABLE 421 Display of **BFD MPLS detail** command

This field...	Displays...
MPLS BFD admin	The global configuration state of MPLS BFD on the device: can be either Enabled or Disabled
Minimum TX interval	Desired Min Tx Interval - the minimum interval, in microseconds, the local system will use when transmitting BFD Control packets. The value zero is reserved.
Minimum RX interval	Required Min Rx Interval - the mimi num interval, in microseconds, between received BFD Control packets that this system is capable of supporting. If this value is zero, the transmitting system does not want the remote system to send any periodic BFD Control packets.
Detection time multiplier	The number of times in a single sequence this device waits to receive a BFD message from the peer before determining that the connection to that peer is not operational.

Chapter

Operations, Administration, and Maintenance (OAM) 60

The following OAM features are supported by PowerConnect B-MLXe Series.

- IEEE 802.1ag Connectivity Fault Management (CFM)
- IEEE 802.1ag Connectivity Fault Management (CFM) for C-VLANs and S-VLANs within an ESI
- IEEE 802.1ag Connectivity Fault Management (CFM) for B-VLANs
- Suport for Sub-second IEEE 802.1ag Timers
- IEEE 802.1ag on VPLS Endpoints
- IEEE 802.1ag over VLL
- MPLS OAM – LSP traceroute
- IEEE 802.3ah EFM-OAM
- Ping
- Ping within a VRF
- Trace Route
- Trace Route within a VRF
- Trace-l2 Protocol
- LSP Ping and Traceroute
- FRR bypass LSPs

Operations, Administration, and Maintenance (OAM) implementation refers to the tools and utilities for installing, monitoring, and troubleshooting the network.

IEEE 802.1ag Connectivity Fault Management (CFM)

Overview

IEEE 802.1ag Connectivity Fault Management (CFM) refers to the ability of a network to monitor the health of a service delivered to customers as opposed to just links or individual bridges.

The IEEE 802.1ag CFM standard specifies protocols, procedures, and managed objects to support transport fault management. This allows for the discovery and verification of the path, through bridges and LANs, taken by frames addressed to and from specified network users and the detection, and isolation of a connectivity fault to a specific bridge or LAN.

Ethernet CFM defines proactive and diagnostic fault localization procedures for point-to-point and multipoint Ethernet Virtual Connections that span one or more links. It operates end-to-end within an Ethernet network.

Ethernet OAM capabilities

Ethernet OAM is able to:

- Monitor the health of links (because providers and customers might not have access to the management layer)
- Check connectivity of ports
- Detect fabric failures
- Provide the building blocks for error localization tools
- Give appropriate scope to customers, providers and operators (hierarchical layering of OAM)
- Avoid security breaches

IEEE 802.1ag purpose

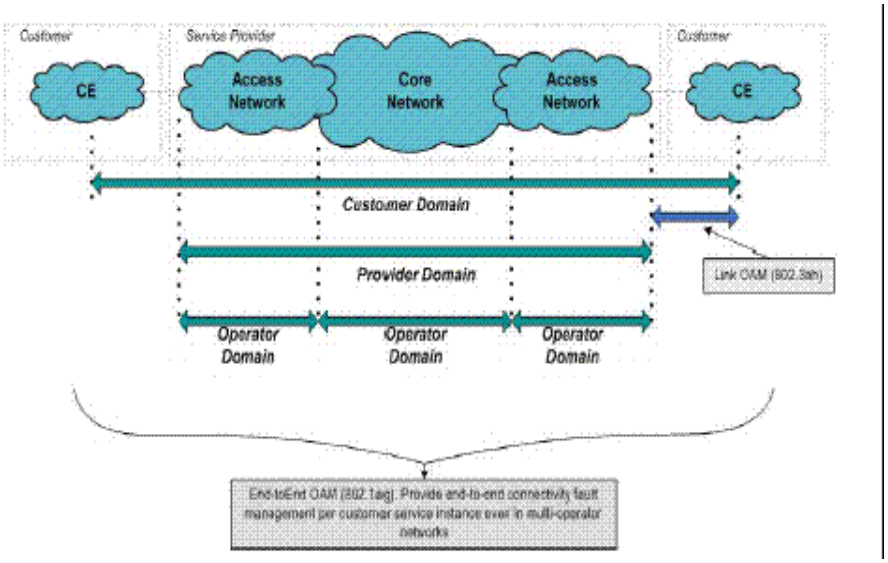
Bridges are increasingly used in networks operated by multiple independent organizations, each with restricted management access to each other's equipment. CFM provides capabilities for detecting, verifying and isolating connectivity failures in such networks.

There are multiple organizations involved in a Metro Ethernet Service: Customers, Service Providers and Operators.

Customers purchase Ethernet Service from Service Providers. Service Providers may utilize their own networks, or the networks of other Operators to provide connectivity for the requested service. Customers themselves may be Service Providers, for example a Customer may be an Internet Service Provider which sells Internet connectivity.

Operators will need minimal Ethernet OAM. Providers will need more comprehensive Ethernet OAM for themselves and to allow customers better monitoring functionality.

FIGURE 234 OAM Ethernet tools



IEEE 802.1ag provides hierarchical network management

Maintenance Domain (MD)

A Maintenance domain is part of a network controlled by a single operator. In Figure 234, we have customer domain, provider domain and operator domain.

Maintenance Domain level (MD level)

The MD levels are carried on all CFM frames to identify different domains.For example, in Figure 234, some bridges belong to multiple domains. Each domain associates a MD level.

- **Customer Level:** 5-7
- **Provider Level:** 3-4
- **Operator Level:** 0-2

Maintenance Association (MA)

Every MD can be further divided into smaller networks having multiple Maintenance End Points (MEP). Usually MA is associated with a service instances (for example a VLAN or a VPLS).

Maintenance End Point (MEP)

MEP is located on the edge of an MA. It defines the endpoint of the MA. Each MEP has unique ID (MEPID) within MA. The connectivity in a MA is defined as connectivity between MEPs. MEP generates Continuity Check Message and multicasts to all other MEPs in same MA to verify the connectivity.

Maintenance Intermediate Point (MIP)

MIP is located within a MA. It responds to Loopback and Linktrace messages for Fault isolation.

Mechanisms of Ethernet IEEE 802.1ag OAM

Mechanisms supported by IEEE 802.1ag include Connectivity Check (CC), Loopback, and Link trace. Connectivity Fault Management allows for end-to-end fault management that is generally reactive (through Loopback and Link trace messages) and connectivity verification that is proactive (through Connectivity Check messages).

Fault detection (continuity check message)

The Continuity Check Message (CCM) provides a means to detect hard and soft faults such as software failure, memory corruption, or misconfiguration. The failure detection is achieved by each Maintenance End Point (MEP) transmitting a CCM periodically within its associated Service Instance.

As a result, MEPs also receive CCMs periodically from other MEPs. If a MEP on local Bridge stops receiving the periodic CCMs from peer MEP on a remote Bridge, it can assume that either the remote Bridge has failed or failure in the continuity of the path has occurred. The Bridge can subsequently notify the network management application about the failure and initiate the fault verification and fault isolation steps either automatically or through operator command.

A CCM requires only N transmissions within its member group, where N is the number of members within the member group. In other words, if a Virtual Bridge LAN Service has N members, only N CCMs need to be transmitted periodically – one from each.

Continuity Check (CC) messages are periodic hello messages multicast by a MEP within the maintenance domain, at the rate of X; X can be 3.3 milliseconds (ms), 10ms, or 100ms, 1 second, 1 minute, or 10 minutes. All Maintenance association Intermediate Points (MIPs) and MEPs in that domain will receive it but will not respond to it. The receiving MEPs will build a MEP database that has entities of the format. MEPs receiving this CC message will catalog it and know that the various maintenance associations (MAs) are functional, including all intermediate MIPs.

CCMs are not directed towards any specific; rather they are multicast across the entire point-to-point or multipoint service on a regular basis. Accordingly, one or more service flows, including the determination of MAC address reachability across a multipoint network, are monitored for connectivity status with IEEE 802.1ag.

Fault verification (Loopback messages)

A unicast Loopback Message is used for fault verification. To verify the connectivity between MEP and its peer MEP or a MIP, the Loopback Message is initiated by a MEP with a destination MAC address set to the MAC address of either a Maintenance association Intermediate Point (MIP) or the peer MEP. The receiving MIP or MEP responds to the Loopback Message with a Loopback Reply.

A Loopback message helps a MEP identify the precise fault location along a given MA. A Loopback message is issued by a MEP to a given MIP along an MA. The appropriate MIP in front of the fault will respond with a Loopback reply. The MIP behind the fault will not respond. For Loopback to work, the MEP must know the MAC address of the MIP to ping.

Fault isolation (Linktrace messages)

Linktrace mechanism is used to isolate faults at Ethernet MAC layer. Linktrace can be used to isolate a fault associated with a given Virtual Bridge LAN Service. It should be noted that fault isolation in a connectionless (multi-point) environment is more challenging than a connection oriented (point-to-point) environment. In case of Ethernet, fault isolation can be even more challenging since a MAC address can age out when a fault isolates the MAC address. Consequently a network-isolating fault results in erasure of information needed for locating the fault.

A Linktrace Message uses a set of reserved multicast MAC address. The Linktrace Message gets initiated by a MEP and traverses hop-by-hop and each Maintenance Point (a MEP or MIP) along the path intercepts this Linktrace Message and forwards it onto the next hop after processing it until it reaches the destination MEP. The processing includes looking at the destination MAC address contained in the Linktrace Message.

Each MP along the path returns a unicast Linktrace Reply back to the originating MEP. The MEP sends a single LTM to the next hop along the trace path; however, it can receive many Linktrace Responses from different MPs along the trace path and the destination MEP as the result of the message traversing hop by hop. As mentioned previously, the age-out of MAC addresses can lead to erasure of information at MIPs, where this information is used for the Linktrace mechanism. Possible ways to address this behavior include:

- Carrying out Linktrace following fault detection or verification such that it gets exercised within the window of age-out.
- Maintaining information about the destination MEP at the MIPs along the path using CCMs.
- Maintaining visibility of path at the source MEPs through periodic LTMs

Linktrace may also be used when no faults are apparent in order to discover the routes normally taken by data through the network. In the rare instances during network malfunctions where Linktrace cannot provide the information needed to isolate a fault, issuing Loopback Messages to MPs along the normal data path may provide additional useful information.

The Linktrace message is used by one MEP to trace the path to another MEP or MIP in the same domain. It is needed for Loopback (Ping). All intermediate MIPs respond back with a Link trace reply to the originating MEP. After decreasing the TTL by one, intermediate MIPs forward the Link trace message until the destination MIP or MEP is reached. If the destination is a MEP, every MIP along a given MA responds to the originating MEP. The originating MEP can then determine the MAC address of all MIPs along the MA and their precise location with respect to the originating MEP.

Configuring IEEE 802.1ag CFM

Enabling or disabling CFM

To enable or disable the CFM protocol globally on the devices and enter into the CFM Protocol Configuration mode, enter a command such as the following.

```
NetIron#(config)cfm-enable
NetIron(config-cfm)#
```

Syntax: [no] cfm-enable

The **no** form of the command disables the CFM protocol.

Creating a Maintenance Domain

A Maintenance Domain is the network or the part of the network for which faults in connectivity are to be managed. A Maintenance Domain consists of a set of Domain Service Access Points.

A Maintenance Domain is, or is intended to be, fully connected internally. A Domain Service Access Point associated with a Maintenance Domain has connectivity to every other Domain Service Access Point in the Maintenance Domain, in the absence of faults.

Each Maintenance Domain can be separately administered.

The **domain-name** command in CFM protocol configuration mode creates a maintenance domain with a specified level and name and enters the Specific Maintenance Domain mode specified in the command argument.

```
NetIron(config-cfm)#domain-name VPLS-SP level 4
NetIron(config-cfm-md-VPLS-SP)#
```

Syntax: [no] domain-name NAME [id <*md-id*>] [level <*level*>]

The **NAME** parameter specifies the domain name. The NAME attribute is case-sensitive.

The **id** <*md-id*> is the Maintenance Domain Index. It is an optional parameter. The range is 1 - 4090.

The **level** parameter sets the domain level in the range 0 – 7. When the domain already exists, the **level** argument is optional. The levels are.

Customer's Domain Levels: 5 - 7

Provider Domain Levels: 3 - 4

Operator Domain Levels: 0 – 2

The **no** form of the command removes the specified domain from the CFM Protocol Configuration mode.

Setting Maintenance Domain parameters

Creating Maintenance Associations

The Maintenance Association Identifier is unique over the domain. If the Maintenance Association Identifier is globally unique, then that domain is global. CFM can detect connectivity errors only for a list of MEPs with unique MAIDs.

The **ma-name** command, in Maintenance Domain mode, creates a maintenance association within a specified domain. The **ma-name** command changes the Maintenance Domain mode to a Specific Maintenance Association mode.

```
NetIron(config-cfm-md-VPLS-SP)# ma-name ma_1 vlan-id 30 priority 4
NetIron(config-cfm-md-VPLS-SP-ma-ma_1)#
```

Syntax: [no] ma-name NAME [id <*ma-id*>] [esi <*esi-id*>] [vlan-id <*vlan-id*>][vpls-id <*vpls-id*>][priority <*priority*>]

The **NAME** parameter specifies the maintenance association name. The NAME attribute is case-sensitive.

The **id** <*ma-id*> is the Maintenence Association Index. It is an optional parameter. The range is 1 - 4090.

The **esi-id** specifies a unique ESI identifier of the maintenance association. In case of creating a MA a ESI ID should be set. This option is available only on platforms that support the Ethernet Service Instance (ESI) framework.

The **vlan-id** specifies a unique VLAN identifier of the maintenance association in the range <*1-4090*>. In case of creating a MA a VLAN ID should be set.

The **vpls-id** specifies a unique VPLS identifier of the maintenance association. In case of creating a MA, a VPLS ID should be set.

The **priority** parameter specifies the priority of the CCM messages, sent by MEPs, in the range <*0-7*>. When the maintenance association is already created, the **priority** argument is optional.

The **no** form of the command removes the created MA.

Configuring a CCM interval for a Maintenance Association (MA)

The **ccm-interval** command sets the time interval between two successive Continuity Check messages (CCMs) that are sent by MEPs in the specified Maintenance Domain. The default value is 10 seconds.

```
NetIron(config-cfm)#domain name VPLS-SP level 4
NetIron(config-cfm-md-VPLS-SP)#ma-name ma_1 vlan-id 30 priority 3
NetIron(config-cfm-md-VPLS-SP-ma-ma_1)#ccm-interval 10-second
NetIron(config-cfm-md-VPLS-SP-ma-ma_1)#
```

Syntax: [no] ccm-interval [1-second | 1-minute | 10-second | 10-minute|3.3-ms | 10-ms | 100-ms]

The **1 second** parameter sets the time interval between two successive CCM packets to 1 second.

The**1 minute** parameter sets the time interval between two successive CCM packets to 1 minute.

The **10 second** parameter sets the time interval between two successive CCM packets to 10 seconds.

The **10 minute** parameter sets the time interval between two successive CCM packets to 10 minutes.

The **3.3 milliseconds** parameter sets the time interval between two successive CCM packets to 3.3 milliseconds.

The **10 milliseconds** parameter sets the time interval between two successive CCM packets to 10 milliseconds.

The **100 milliseconds** parameter sets the time interval between two successive CCM packets to 100 milliseconds.

Configuring local ports

The **mep** command, in Maintenance Association mode, adds local ports as MEP to a specific maintenance association. If configuring a CFM packet to a “down” MEP, it will need to be sent out on the port on which it was configured. If configuring a CFM packet to an “up” MEP, it will need to be sent to the entire VLAN for multicast traffic, and unicast traffic will need to be sent to a particular port according to the MAC table.

Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity:

- The list of MEPs configured with identical values for MA ID defines an MA.
- Each Bridge has its own Maintenance Association managed object for an MA.
- Each individual MEP is configured with a ID that is unique within that MA.
- Each MEP is associated with a Service Access Point that provides access to a single service instance.

NOTE
When configuring 802.1ag over VPLS, if the VPLS endpoint is deleted from the configuration, the MEP configuration is deleted under CFM without warning.

To add local ports to an upstream MEP, enter commands such as the following.

```
NetIron(config-cfm)# domain name VPLS-SP level 4
NetIron(config-cfm-md-VPLS-SP)# ma-name ma_1 vlan-id 30 priority 3
NetIron(config-cfm-md-VPLS-SP-ma_1)# mep 1 up port eth 2/1
NetIron(config-cfm-md-VPLS-SP-ma_1)#
```

Syntax: [no] **mep** <*mep-id*> [**up** I **down**] [**vlan** <*vlan-id*> **port ethernet** <*slot/port*> | **port ethernet** <*slot/port*>]

The **mep-id** parameter specifies the maintenance end point ID (mandatory) in the range <*1-8192*>.

The **up** parameter sets the MEP direction away from the monitored VLAN.

The **down** parameter sets the MEP direction towards the monitored VLAN.

The **vlan-id** parameter specifies the VLAN end-points. It is configured only for MAs associated with VPLS and not configured for MAs with a VLAN.

The **port-id** parameter specifies the target interface on which it is used.

The **no** form of the command removes the specified MEPs.

Configuring Remote MEPs

The **remote-mep** command is used to configure the remote MEP's you are expecting. If a remote MEP is not specified, the remote MEP database is built based on the CCM. If one remote MEP never sends CCM, the failure can not be detected.

```
NetIron(config-cfm-md-VPLS-SP)# ma-name ma_1 vlan-id 30
NetIron(config-cfm-md-VPLS-SP-ma_1)# remote-mep 1 to 120
NetIron(config-cfm-md-VPLS-SP-ma_1)#
```

Syntax: [no] **remote-mep** <*mep-id*> [**to** <*mep-id*>]

The **mep-id** parameter specifies the maintenance end point ID (mandatory) in the range <*1-8192*>.

The **no** form of the command removes the specified remote MEPs.

Setting the Remote Check Start-Delay

When configuring the remote MEPs range, you can set a wait time before the MEPs come up and the CCM check operation is started. The default is set to 30 seconds.

```
NetIron#(config)cfm-enable
NetIron(config-cfm)#rmep-check start-delay 120
NetIron(config-cfm)#
```

Syntax: [no] **rmep-check start-delay** < *seconds*>

The **seconds** parameter is the wait time interval before the CCM check is started. The range is 10 – 600 seconds.

Specifying MIP creation policy

The **mip-policy** command, in Maintenance Association mode, specifies the conditions in which MIPs are automatically created on ports.

NOTE
MIP functionality of 802.1ag over VPLS with sub-second timer will have all the configuration restrictions of the VPLS CPU-protection.

A MIP can be created on a port and VLAN, only when explicit or default policy has been defined for them. For a specific port and VLAN a MIP will be created at the lowest of the levels. Additionally, the level created should be the next higher than the MEP level defined for these port and VLAN.

```
NetIron(config-cfm)#domain name VPLS-SP level 4
NetIron(config-cfm-md-VPLS-SP)#ma-name ma_1 vlan-id 30
NetIron(config-cfm-md-VPLS-SP-ma_1)#mip-policy explicit
NetIron(config-cfm-md-VPLS-SP-ma_1)#
```

Syntax: [no] mip-policy [explicit |default]

Use the **explicit** parameter to specify that explicit MIPs are configured only if a MEP exists on a lower MD Level.

Use the **default** parameter to specify that MIPs will always be created.

The **no** form of the command restores the default Policy.

Y.1731 performance management

The Y.1731 feature provides the following performance monitoring capability for point-to-point links as defined in ITU-T Rec Y.1731:

- Frame Delay Measurement (ETH-DM)
- Frame Delay Measurement Variation

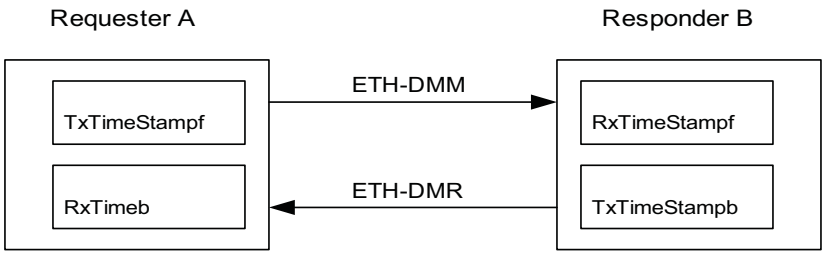
When using Y.1731, ETH-DM can be performed using one-way ETH-DM and two-way ETH-DM.

NOTE
This release of the Multi-Service IronWare only supports two-way ETH-DM.

About Y.1731

Table 422 shows an ETH-DM requester and responder.

TABLE 422 ETH-DM requester and responder.



ETH-DM packets are transmitted, received and processed by LP CPU and are timestamped by the hardware on transmit and receive path.

An ETH-DM packet contains 4 timestamps for measuring the round-trip delay.

Requester A, transmits ETH-DMM packets with TxTimeStampf (timestamp at the transmission time of the packet).

Responder B, responds with an ETH-DMR packet using two timestamps to account for its processing time: RxTimeStampf (Timestamp at the time of receiving the DMM packet) and TxTimeStampb (timestamp at the time of transmitting the DMR packet).

Upon receiving an ETH-DMR packet, requester A stamps the packet with RxTimeb (timestamp at the time the DMR packet is received).

Frame Delay = (RxTimeb – TxTimeStampf) – (TxTimeStampb – RxTimeStampf)

This release provides Y.1731 support for the following:

- VLANs
- VPLS
  - Both VC-mode tagged and raw
- VLL
  - Both tagged and raw modes
- Up and Down MEPs for VLANs, VPLS, and VLL

- Over LAG ports
  - The active primary port of the trunk would be used to transmit ETH-DM frames in case of down MEP
- Through 802.1ag MIPs
  - MIP would behave as a transient node for ETH-DM frames

Configuration considerations:

When using Y.1731, consider the following:

- ETH-DM is reliable only if the transmitted DM frame (DMM) and received DM reply (DMR) are on the same line processor (LP). In the event that they are different, results will not be accurate.
- Maximum frame-delay that can be measured is 4 seconds. If a DMR packet is received with a delay greater than 4 seconds, the packet is discarded and ignored.
- ETH-DM does not gather path data. To determine which path the DM applies to, use the **cfm linktrace domain** command because ETH-DM frames follow the same path.
- One-way ETH-DM is not supported in this release of the Multi-Service IronWare.

Configuring Y.1731 performance monitoring

Use the **cfm delay_measurement domain** command to issue the delay measurement. If the number of delay measurement frame is greater than 16, then the last 16 delay measurement replies are printed.

You can issue the **cfm delay measurement** command from different sessions if they are for different **src-meps**. However, if it is for same **src-mep**, it only completes one session at a time.

```
NetIron# cfm delay_measurement domain md2 ma ma2 src-mep 3 target-mep 2
Y1731: Sending 10  delay_measurement to 0012.f2f7.3931, timeout 1000 msec
Type Control-c to abort
Reply from 0012.f2f7.3931: time= 32.131 us
Reply from 0012.f2f7.3931: time= 31.637 us
Reply from 0012.f2f7.3931: time= 32.566 us
Reply from 0012.f2f7.3931: time= 34.052 us
Reply from 0012.f2f7.3931: time= 33.376 us
Reply from 0012.f2f7.3931: time= 31.501 us
Reply from 0012.f2f7.3931: time= 33.016 us
Reply from 0012.f2f7.3931: time= 32.537 us
Reply from 0012.f2f7.3931: time= 32.492 us
Reply from 0012.f2f7.3931: time= 32.552 us
sent = 10 number = 10 A total of 10 delay measurement replies received.
Success rate is 100 percent (10/10)

================================================================================
Round Trip Frame Delay Time      : min =  31.501 us  avg =  32.586 us  max =  34.052 us

Round Trip Frame Delay Variation : min =      45 ns  avg =     839 ns  max =   1.875 us
================================================================================
```

Syntax: **cfm delay_measurement domain** <domain-name> **ma** <ma-name> **src-mep** <mep-id> **dst-mep** <mep-id> [**timeout** <timeout>] [**number** <number>]

The **domain** <domain-name> parameter specifies the maintenance domain to be used for a delay measurement message. The <domain-name> attribute is case-sensitive.

The **ma** <ma-name> parameter specifies the maintenance association to be used for a delay measurement message. The <ma-name> attribute is case-sensitive.

The **src-mep** <mep-id> parameter specifies the source mep-id in the range <1-8192>.

The **dst-mep** <mep-id> parameter specifies the destination mep-id in the range <1-8192>.

The **number** <number> parameter specifies the number of delay_measurement messages to be sent. The range is 1-1000. The default value is 10. This is an optional parameter.

The **timeout** <timeout> parameter specifies the timeout used to wait for previous delay_measurement reply before sending the next delay_measurement message. The range is 1-4 seconds. The default value is 1second. This is an optional parameter.

If a **delay_measurement** reply is received before the timeout, then the next delay measurement frame is sent immediately after processing the delay measurement reply. However, if the **delay measurement** reply is not received within the specified timeout, then the next **delay measurement** frame will be sent.

Y. 1731 show commands

Use the **show cfm statistic delay_measurement domain** command to display delay measurement statistics. If the command is issued gain, the output is replaced with the new values.

```
NetIron#show cfm statistics delay_measurement domain md2 ma ma2 rmep-id 2
Domain: md2 Level: 7
Maintenance association: ma2 VLAN ID: 2 Priority: 7

================================================================================================
Round Trip Frame Delay Time : min =  31.501 us  avg =  32.586 us  max =  34.052 us

Round Trip Frame Delay Variation : min = 45 ns     avg =     839 ns     max =   1.875 us
================================================================================================

Port Used to transmit delay_measurement: 2/2
Number of delay_measurement frames Used to calculate Statistics: 10
```

Syntax: show cfm statistics delay_measurement domain <domain- name> **ma** <ma-name> **rmep** <rmep-id>

The **domain** <domain-name> parameter specifies the maintenance domain to be used for a delay measurement message. The <domain-name> attribute is case-sensitive.

The **ma** <ma-name> parameter specifies the maintenance association to be used for a delay measurement message. The <ma-name> attribute is case-sensitive.

The **rmep** <rmep-id> parameter specifies the remote mep id to be used for a delay measurement message.

Sample configuration

1. MEP configuration (prerequisite for ETH-DM to work).

Requester-A :

```
NetIron(config)#cfm-enable
NetIron(config-cfm)# domain-name md2 level 7
NetIron(config-cfm-md-md2)#  ma-name ma2 vlan-id 2 priority 7
NetIron(config-cfm-md-md2-ma-ma2)# mep 3 down port ethe 2/2
NetIron(config-cfm-md-md2-ma-ma2)#

Responder-B:
NetIron(config)#cfm-enable
NetIron(config-cfm)# domain-name md2 level 7
NetIron(config-cfm-md-md2)#  ma-name ma2 vlan-id 2 priority 7
NetIron(config-cfm-md-md2-ma-ma2)# mep 2 down port ethe 2/2
NetIron(config-cfm-md-md2-ma-ma2)#
```

2. Issue the **cfm delay_measurement** command.

```
NetIron# cfm delay_measurement domain md2 ma ma2 src-mep 3 target-mep 2
Y1731: Sending 10  delay_measurement to 0012.f2f7.3931, timeout 1000 msec
Type Control-c to abort
Reply from 0012.f2f7.3931: time= 32.131 us
Reply from 0012.f2f7.3931: time= 31.637 us
Reply from 0012.f2f7.3931: time= 32.566 us
Reply from 0012.f2f7.3931: time= 34.052 us
Reply from 0012.f2f7.3931: time= 33.376 us
Reply from 0012.f2f7.3931: time= 31.501 us
Reply from 0012.f2f7.3931: time= 33.016 us
Reply from 0012.f2f7.3931: time= 32.537 us
Reply from 0012.f2f7.3931: time= 32.492 us
Reply from 0012.f2f7.3931: time= 32.552 us
sent = 10 number = 10 A total of 10 delay measurement replies received.
Success rate is 100 percent (10/10)

================================================================================
Round Trip Frame Delay Time      : min =  31.501 us  avg =  32.586 us  max =  34.052 us

Round Trip Frame Delay Variation : min =      45 ns  avg =     839 ns  max =   1.875 us
================================================================================
```

3. Issue the **show cfm statistic delay_measurement domain** command.

```
NetIron#show cfm statistics delay_measurement domain md2 ma ma2 rmep-id 2
Domain: md2 Level: 7
Maintenance association: ma2 VLAN ID: 2 Priority: 7

================================================================================
Round Trip Frame Delay Time : min =  31.501 us  avg =  32.586 us  max =  34.052 us

Round Trip Frame Delay Variation : min = 45 ns     avg =     839 ns     max =   1.875 us
================================================================================

Port Used to transmit delay_measurement: 2/2
Number of delay_measurement frames Used to calculate Statistics: 10
```

CFM monitoring and show commands

Sending linktrace messages

The **cfm linktrace domain** command sends a linktrace message to a specified MEP in the domain. Enter a command such as the following to send a linktrace mesage to a specified MEP in the domain.

```
NetIron# cfm linktrace domain VPLS-SP ma ma_1 src-mep 21 target-mep 1 timeout 10 t
Linktrace to 000c.dbfb.5378 on Domain VPLS-SP, level 4: timeout 10ms, 4 hops
----------------------------------------------------------------
Hops           MAC             Ingress     Ingress Action   Relay Action
            Forwarded          Egress      Egress Action    Nexthop
----------------------------------------------------------------
   1    000c.dbe2.6ea0                                        RLY_FDB
           Forwarded               5/4           EgrOK
   2    000c.dbfb.5378              7/2           IgrOK      RLY_HIT
        Not Forwarded
Destination 000c.dbfb.5378 reached
```

Syntax: **[no] cfm linktrace domain NAME ma MA-NAME src- mep** <*mep-id*> **dst-mip HH:HH:HH:HH:HH:HH | target-mep** <*mep-id*>} **[timeout** <*timeout*>**] [ttl** <*TTL*>**]**

The **domain** NAME parameter specifies the maintenance domain to be used for a linktrace message. The NAME attribute is case-sensitive.

The **ma** MA NAME parameter specifies the maintenance association to be used for a linktrace message. The NAME attribute is case-sensitive.

The **src-mep** <*mep-id*> parameter specifies the Source ID in the range 1 – 8192.

The **dst-mip** HH:HH:HH:HH:HH:HH parameter specifies the MAC-address of the MIP linktrace destination.

The **dst-mep** <*mepid*> parameter specifies the Destination ID of the linktrace destination.

The **timeout** <*timeout*> parameter specifies the time to wait for a linktrace reply. The range is 1 – 30 seconds.

The **ttl** <*TTL*> parameter specifies the initial TTL field value in the range 1 – 64.The default is 8 seconds.

Sending loopback messages

The **cfm loopback domain** command, sends a loopback message to a specific MIP in a specified domain.

```
NetIron#cfm loopback domain VPLS-SP ma ma_1 src-mep 2 target-mep 1 timeout 10
number 10
cfm: Sending 10  Loopback to 000c.dbfb.5378, timeout 10 msec
Type Control-c to abort
Reply from 000c.dbfb.5378: time=1ms
Reply from 000c.dbfb.5378: time<1ms
Reply from 000c.dbfb.5378: time<1ms
Reply from 000c.dbfb.5378: time<1ms
Reply from 000c.dbfb.5378: time<1ms
Reply from 000c.dbfb.5378: time<1ms
```

```
Reply from 000c.dbfb.5378: time<1ms
Reply from 000c.dbfb.5378: time<1ms
Reply from 000c.dbfb.5378: time<1ms
Reply from 000c.dbfb.5378: time<1ms
A total of 10 loopback replies received.
Success rate is 100 percent (10/10), round-trip min/avg/max=0/0/1 ms.
```

Syntax: [no] cfm loopback domain NAME ma MA-NAME scr-mep <*mep-id*> {dst-mip HH:HH:HH:HH:HH:HH | target-mep <*mep-id*>} [number <*number*>] [timeout <*timeout*>]

The **domain** NAME parameter specifies the maintenance domain to be used for a linktrace message. The NAME attribute is case-sensitive.

The **ma** MA-NAME parameter specifies the maintenance association to be used for a linktrace message. The MA-NAME attribute is case-sensitive.

The **src-mep** <*mep-id*> parameter specifies the Source ID in the range <*1-8192*>.

The **dst- mip** HH:HH:HH:HH:HH:HH parameter specifies the MAC address of the MIP linktrace destination.

The **dst-mep** <*mep-id*> parameter specifies the Destination ID in the range <*1-8192*>.

The **number** <*number*> parameter specifies the number of loopback messages to be sent.

The **timeout** <*timeout*> parameter specifies the timeout used to wait for linktrace reply.

Displaying CFM configurations

The **show cfm** command, displays the current configuration and status of CFM. For the **show cfm** command to take effect, CFM should first be enabled in Protocol Configuration mode.

```
NetIron#show cfm
Domain: md2
Index: 1
Level: 6
  Maintenance association: ma2
  Ma Index: 1
  CCM interval: 10000 ms
  VLAN ID: 2
  Priority: 6
  MEP   Direction  MAC             PORT
  ====  =========  =========       ====
     3  DOWN       0012.f2f7.3831  ethe 2/2
```

Syntax: show cfm [domain NAME] [ma NAME]

The [**domain NAME**] parameter specifies a domain for display. By default, all defined domains are shown.

The [**ma NAME**] parameter specifies the maintenance association name. By default, all defined domains are shown.

TABLE 423 Show CFM output descriptions

This field...	Displays...
Domain	The Domain is the network or the part of the network for which faults in connectivity are displayed.
Index	The Domain Index.
Level	The level is the domain level in the range <*0-7*>. The levels can be: • Customer's MD levels: 5 - 7 • Provider's MD levels: 3 - 4 • Operator's MD levels: 0 – 2
Maintenance Association	The maintenance association name.
Ma Index	The Maintenance Association Index.
CCM interval	The time interval between two successive Continuity Check messages (CCMs) that are sent by MEPs in the specified Maintenance Domain.
VLAN ID	The VLAN identifier of the maintenance association.
VPLS ID	The VPLS identifier of the maintenance association.
Priority	The priority of the CCM messages, sent by 'MEPs, in the range <*0-7*>.
MEP	The maintenance end point ID
Direction	DIsplays the direction the MEP was sent: Up - The MEP direction away from the monitored VLAN. Down - The MEP direction is towards the monitored VLAN.
MAC	DIsplays the associated MAC Address.
PORT	DIsplays the associated port.

TABLE 423 Show CFM output descriptions

This field...	Displays...
MIP	DIsplays the associated MIP
VLAN	DIsplays the associated VLAN.

The show cfm brief show a summary of the configured MEPs and RMEPs.

```
NetIron MLXe-4000 Router#show cfm brief
Domain: md2
Index: 1
Level: 6  Num of MA: 1
  Maintenance association: ma2
  MA Index: 1
  CCM interval: 10000 ms
  VLAN ID: 2
  Priority: 6
  Num of MEP: 1               Num of RMEP: 1
  rmepstart: 0 rmepfail: 0 rmepok 1
```

Syntax: show cfm [domain NAME] [ma NAME] brief

TABLE 424 Show cfm brief output description

This field...	Displays...
Domain	The Domain is the network or the part of the network for which faults in connectivity are displayed.
Index	The Domain Index.
Level	The level is the domain level in the range <*0-7*>. The levels can be: • Customer's MD levels: 5 - 7 • Provider's MD levels: 3 - 4 • Operator's MD levels: 0 – 2
Maintenance Association	The maintenance association name.
Ma Index	The Maintenance Association Index.
CCM interval	The time interval between two successive Continuity Check messages (CCMs) that are sent by MEPs in the specified Maintenance Domain.
VLAN ID	The VLAN identifier of the maintenance association.
VPLS ID	The VPLS identifier of the maintenance association.
Priority	The priority of the CCM messages, sent by 'MEPs, in the range <*0-7*>.
Numof MEP	The number of MEPs configured.
Num of RMEP	The number of remote MEPs configured
remepstart	The number of RMEPs in the start state.
rmepfail	The number of RMEPs that have failed.
rmepok	The number of RMEPs in an OK state.

Displaying connectivity statistics

The **show cfm connectivity** command, displays connectivity statistics for the remote database. For the **show cfm connectivity** command to take effect, CFM should first be enabled in the Protocol Configuration mode.

```
NetIron#show cfm connectivity
Domain: md2 Index: 1
Level: 6
  Maintenance association: ma2
  MA Index: 1
  CCM interval: 10000 ms
  VLAN ID: 2
  Priority: 6
  RMEP  MAC             VLAN/PEER         AGE     PORT     SLOTS
  ====  =====           ===============  =======  ======== =========
     2  0012.f2f7.3931        2               20       2/2 2
```

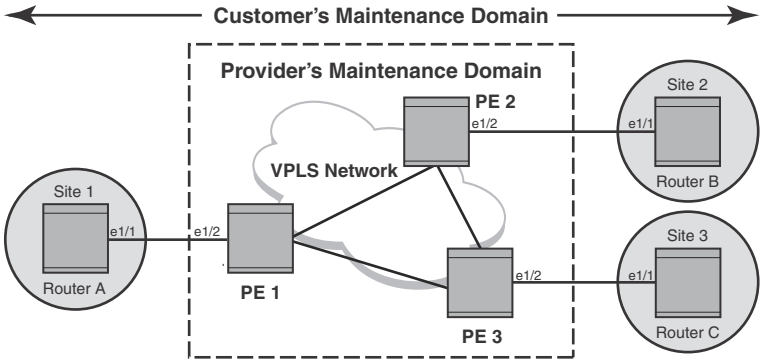
Syntax: show cfm connectivity

TABLE 425 Show CFM connectivity output descriptions

This field...	Displays...
Domain	The Domain is the network or the part of the network for which faults in connectivity are displayed.
Index	The Domain Index.
Level	The level is the domain level in the range <*0-7*>. The levels can be: • Customer's MD levels: 5 - 7 • Provider's MD levels: 3 - 4 • Operator's MD levels: 0 - 2
Maintenance association	The maintenance association name.
Ma Index	The Maintenance Association Index.
CCM interval	The time interval between two successive Continuity Check messages (CCMs) that are sent by MEPs in the specified Maintenance Domain.
VPLS ID	The VPLS identifier of the maintenance association.
Priority	The priority of the CCM messages, sent by MEPs, in the range <*0-7*>.
RMEP	The remote maintenance end point ID
MAC	DIsplays the associated MAC Address.
VLAN or VC	VLAN ID or VC label learned from the CCM packet. VC label is in hexadecimal format.
Age	Uptime since RMEP discovery or from last age out
PORT	Displays the associated port.
SLOTMASK	Mask of slots that are receiving CCM packets which are used for multi-slot trunks. For example a value of 0005 indicates Slots 1 and 3.

Sample configuration for a customer's domain

FIGURE 235 Sample configuration



Configuring Router A

CFM configuration steps for Router A are listed below.

1. To enable CFM, enter the following command.

   ```
   RouterA(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

   ```
   RouterA(config-cfm)#domain-name CUST_1 level 7
   ```

3. Create a maintenance association within a specified domain of **vlan-id 30** with a priority 3 .

   ```
   RouterA(config-cfm-md-CUST_1)#ma-name ma_5 vlan-id 30 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages to **10-seconds**.

   ```
   RouterA(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
   ```

5. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port 1/2 to a specified maintenance association.

   ```
   RouterA(config-cfm-md-CUST_1-ma-ma_5)#mep 2 down vlan 30 port ethe 1/2
   ```

Configuring Router B

CFM configuration steps for Router B are listed below.

1. To enable CFM for VPLS, enter the following command.

   ```
   RouterB(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name **CUST_1** and level **7**.

   ```
   RouterB(config-cfm)#domain-name CUST_1 level 7
   ```

3. Create a maintenance association within a specified domain of **vlan-id 30** with a priority **5** .

   ```
   RouterB(config-cfm-CUST_1)#ma-name ma_5 vlan-id 30 priority 5
   ```

4. Set the time interval between successive Continuity Check Messages to **10-seconds**.

   ```
   RouterB(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
   ```

5. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port **1/2** as MEP to a specified maintenance association.

```
RouterB(config-cfm-md-CUST_1-ma-ma_5)#mep 2 down vlan 30 port ethe 1/2
```

Configuring Router C

CFM configuration steps for Router C are listed below.

1. To enable CFM for VPLS, enter the following command.

```
RouterC(config)#cfm-enable
```

2. Create a maintenance domain with a specified name **CUST_1** and level **7**.

```
RouterC(config-cfm)#domain-name CUST_1 level 7
```

3. Create a maintenance association within a specified domain of **vlan-id 30** with a priority **4** .

```
RouterC(config-cfm-CUST_1)#ma-name ma_5 vlan-id 30 priority 4
```

4. Set the time interval between successive Continuity Check Messages to **10-seconds**.

```
RouterC(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```

5. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port **1/2** as MEP to a specified maintenance association.

```
Routerc(config-cfm-md-CUST_1-ma-ma_5)#mep 1 down vlan 30 port ethe 1/2
```

Configuring CFM using Provider Bridges

Below is an example for configuring CFM when using Provider Bridges configurations as in the figure on "Sample configuration" on page 2189.

Configuring Router A

CFM configuration steps for Router A are listed below.

1. Create the port-based VLAN that contains the tagged interface that you want to use by entering the following commands.

```
RouterA(config)# vlan30
RouterA(config-vlan-30)# tagged ethe 1/1
```

2. To enable CFM, enter the following command.

```
RouterA(config)#cfm-enable
```

3. Create a maintenance domain with a specified name CUST_1 and level 7.

```
RouterA(config-cfm)#domain-name CUST_1 level 7
```

4. Create a maintenance association within a specified ESI **Site1vlan30**, and a **vlan-id 30** with a priority **3**.

```
RouterA(config-cfm-md-CUST_1)#ma-name ma_5 esi Site1vlan30 vlan-id 30 priority
3
```

5. Set the time interval between successive Continuity Check Messages to **10-seconds**.

```
RouterA(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```

6. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port 1/2 to a specified maintenance association.

```
RouterA(config-cfm-md-CUST_1-ma-ma_5)#mep 2 down port ethe 1/2
```

7. To configure the hostnameas **RouterA**, enter a command such as the following.

```
RouterA(config)#hostname RouterA
```

8. Configure interface ethernet 1/1 as the custome-edge by entering the following commands.

```
RouterA(config)#interface ethernet 1/1
RouterA(config-if-e10000-1/1)#port-type customer-edge
RouterA(config-if-e10000-1/1)enable
RouterA(config-if-e10000-1/1)end
```

Configuring Router B

CFM configuration steps for Router B are listed below.

1. Create the port-based VLAN that contains the tagged interface that you want to use by entering the following commands.

```
RouterB(config)# vlan30
RouterB(config-vlan-30)# tagged ethe 1/1
```

2. To enable CFM, enter the following command.

```
RouterB(config)#cfm-enable
```

3. Create a maintenance domain with a specified name CUST_1 and level 7.

```
RouterB(config-cfm)#domain-name CUST_1 level 7
```

4. Create a maintenance association within a specified ESI **Site2vlan30**, and a **vlan-id 30** with a priority **3** .

```
RouterB(config-cfm-md-CUST_1)#ma-name ma_5 esi Site2vlan30 vlan-id 30 priority
3
```

5. Set the time interval between successive Continuity Check Messages to **10-seconds**.

```
RouterB(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```

6. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port 1/2 to a specified maintenance association.

```
RouterB(config-cfm-md-CUST_1-ma-ma_5)#mep 2 down port ethe 1/2
```

7. To configure the hostnameas **RouterB**, enter a command such as the following.

```
RouterB(config)#hostname RouterB
```

8. Configure interface ethernet 1/1 as the custome-edge by entering the following commands.

```
RouterB(config)#interface ethernet 1/1
RouterB(config-if-e10000-1/1)#port-type customer-edge
RouterB(config-if-e10000-1/1)enable
RouterB(config-if-e10000-1/1)end
```

Configuring Router C

CFM configuration steps for Router C are listed below.

1. Create the port-based VLAN that contains the tagged interface that you want to use by entering the following commands.

```
RouterC(config)# vlan30
RouterC(config-vlan-30)# tagged ethe 1/1
```

2. To enable CFM, enter the following command.

```
RouterC(config)#cfm-enable
```

3. Create a maintenance domain with a specified name CUST_1 and level 7.

```
RouterC(config-cfm)#domain-name CUST_1 level 7
```

4. Create a maintenance association within a specified ESI **Site3vlan30**, and a vlan-id 30 with a priority **3** .

```
RouterC(config-cfm-md-CUST_1)#ma-name ma_5 esi Site3vlan30 vlan-id 30 priority
3
```

5. Set the time interval between successive Continuity Check Messages to **10-seconds**.

```
RouterC(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```

6. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port 1/2 to a specified maintenance association.

```
RouterC(config-cfm-md-CUST_1-ma-ma_5)#mep 2 down port ethe 1/2
```

7. To configure the hostnameas **RouterC**, enter a command such as the following.

```
RouterC(config)#hostname RouterC
```

8. Configure interface ethernet 1/1 as the **custome-edge** by entering the following commands.

```
RouterC(config)#interface ethernet 1/1
RouterC(config-if-e10000-1/1)#port-type customer-edge
RouterC(config-if-e10000-1/1)enable
RouterC(config-if-e10000-1/1)end
```

Provider Bridge NetIron1

1. Create the port-based VLAN that contains the tagged interface that you want to use by entering the following commands.

```
NetIron1(config)# vlan30
NetIron1(config-vlan-30)# tagged ethe 1/1
```

2. Create the ESI NetIron**1vlan300** as an encapsulated SVLAN with the ESI client **Site1vlan30** by entering the following commands.

```
NetIron1(config)#esi NetIron1vlan300 encapsulation svlan
NetIron1(config)#esi-client Site1vlan30
```

3. Add the port-based **VLAN300** that contains the tagged interfaces that you want to use by entering the following commands.

```
NetIron1(config)# vlan300
NetIron1(config-vlan-300)# tagged ethe 1/1 ethe 1/3
```

4. To enable CFM, enter the following command.

```
NetIron1(config)#cfm-enable
```

5. Create a maintenance domain with a specified name **CUST_1** and level **5**.

```
NetIron1(config-cfm)#domain-name CUST_1 level 5
```

6. Create a maintenance association within a specified ESI **Site1vlan30**, and a **vlan-id 30** with a priority **3** .

```
NetIron1(config-cfm-md-CUST_1)#ma-name ma_5 esi Site1vlan30 vlan-id 30
priority 3
```

7. Set the time interval between successive Continuity Check Messages to **10-seconds**.

```
NetIron1(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```

8. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port **1/2** as MEP to a specified maintenance association.

```
NetIron1(config-cfm-md-CUST_1-ma-ma_5)#mep 4 up port ethe 1/2
```

9. To configure the hostname as NetIron**1**, enter a command such as the following.

```
NetIron1(config)#hostname NetIron1
```

10. Configure interface ethernet **1/1** as the **provider network** by entering the following commands.

```
NetIron1(config)#interface ethernet 1/1
NetIron1(config-if-e10000-1/1)#port-type provider-network
NetIron1(config-if-e10000-1/1)enable
NetIron1(config-if-e10000-1/1)end
```

11. Configure interface ethernet **1/2** as the **customer-edge** by entering the following commands.

```
NetIron1(config)#interface ethernet 1/2
NetIron1(config-if-e10000-1/2)#port-type custonmer-edge
NetIron1(config-if-e10000-1/2)enable
NetIron1(config-if-e10000-1/2)end
```

12. Configure interface ethernet **1/3** as the **provider network** by entering the following commands.

```
NetIron1(config)#interface ethernet 1/3
NetIron1(config-if-e10000-1/3)#port-type provider-network
NetIron1(config-if-e10000-1/3)enable
NetIron1(config-if-e10000-1/3)end
```

Provider Bridge NetIron2

1. Create the port-based **VLAN300** that contains the tagged interfaces that you want to use by entering the following commands.

```
NetIron2(config)# vlan300
NetIron2(config-vlan-300)# tagged ethe 1/1 ethe 1/3
```

2. To enable CFM, enter the following command.

```
NetIron2(config)#cfm-enable
```

3. Create a maintenance domain with a specified name **CUST_1** and level **5**.

```
NetIron2(config-cfm)#domain-name CUST_1 level 5
```

4. Create a maintenance association within a specified ESI **Site2vlan30**, and a **vlan-id 30** with a priority **3**.

```
NetIron2(config-cfm-md-CUST_1)#ma-name ma_5 esi Site2vlan30 vlan-id 30
priority 3
```

5. Set the time interval between successive Continuity Check Messages to **10-seconds**.

```
NetIron2(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```

6. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port **1/2** as MEP to a specified maintenance association.

```
NetIron2(config-cfm-md-CUST_1-ma-ma_5)#mep 5 up port ethe 1/2
```

7. To configure the hostname as NetIron**1**, enter a command such as the following.

```
NetIron2(config)#hostname NetIron1
```

8. Configure interface ethernet **1/1** as the **provider network** by entering the following commands.

```
NetIron2(config)#interface ethernet 1/1
NetIron2(config-if-e10000-1/1)#port-type provider-network
NetIron2(config-if-e10000-1/1)enable
NetIron2(config-if-e10000-1/1)end
```

9. Configure interface ethernet **1/2** as the **customer-edge** by entering the following commands.

```
NetIron2(config)#interface ethernet 1/2
NetIron2(config-if-e10000-1/2)#port-type custonmer-edge
NetIron2(config-if-e10000-1/2)enable
NetIron2(config-if-e10000-1/2)end
```

10. Configure interface ethernet **1/3** as the **provider network** by entering the following commands.

```
NetIron2(config)#interface ethernet 1/3
NetIron2(config-if-e10000-1/3)#port-type provider-network
NetIron2(config-if-e10000-1/3)enable
NetIron2(config-if-e10000-1/3)end
```

Provider Bridge NetIron3

1. Create the port-based VLAN that contains the tagged interface that you want to use by entering the following commands.

```
NetIron3(config)# vlan30
NetIron3(config-vlan-30)# tagged ethe 1/2
```

2. Create the ESI NetIron**3vlan300** as an encapsulated SVLAN with the ESI client **Site3vlan30** by entering the following commands.

```
NetIron3(config)#esi NetIron3vlan300 encapsulation svlan
NetIron3(config)#esi-client Site3vlan30
```

3. Add the port-based **VLAN300** that contains the tagged interfaces that you want to use by entering the following commands.

```
NetIron3(config)# vlan300
NetIron3(config-vlan-300)# tagged ethe 1/1 ethe 1/3
```

4. To enable CFM, enter the following command.

```
NetIron3(config)#cfm-enable
```

5. Create a maintenance domain with a specified name **CUST_1** and level **5**.

```
NetIron3(config-cfm)#domain-name CUST_1 level 5
```

6. Create a maintenance association within a specified ESI **Site3vlan30**, and a **vlan-id 30** with a priority **3**.

```
NetIron3(config-cfm-md-CUST_1)#ma-name ma_5 esi Site3vlan30 vlan-id 30
priority 3
```

7. Set the time interval between successive Continuity Check Messages to **10-seconds**.

```
NetIron3(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```

8. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port **1/2** as MEP to a specified maintenance association.

```
NetIron3(config-cfm-md-CUST_1-ma-ma_5)#mep 6 up port ethe 1/2
```

9. To configure the hostname as NetIron**3**, enter a command such as the following.

```
NetIron3(config)#hostname NetIron3
Configure interface ethernet 1/1 as the provider network by entering the
following commands:
NetIron3(config)#interface ethernet 1/1
NetIron3(config-if-e10000-1/1)#port-type provider-network
NetIron3(config-if-e10000-1/1)enable
NetIron3(config-if-e10000-1/1)end
```

10. Configure interface ethernet **1/2** as the **customer-edge** by entering the following commands.

```
NetIron3(config)#interface ethernet 1/2
NetIron3(config-if-e10000-1/2)#port-type custonmer-edge
NetIron3(config-if-e10000-1/2)enable
NetIron3(config-if-e10000-1/2)end
```

11. Configure interface ethernet **1/3** as the **provider network** by entering the following commands.

```
NetIron3(config)#interface ethernet 1/3
NetIron3(config-if-e10000-1/3)#port-type provider-network
NetIron3(config-if-e10000-1/3)enable
NetIron3(config-if-e10000-1/3)end
```

Displaying the connectivity status in a customer's domain

The following output are for 3 VPLS CEs. The 3 CEs are monitoring Ethernet LAN service in VLAN 30. The Ethernet SP is providing transport service for the customer's VLAN 30 through VPLS which is transparent to customer. The customer is only concerned about RMEPs from remote sites.

```
RouterA#show cfm connectivity
Domain: CUST_1 Level: 7
  Maintenance association: ma_5
  CCM interval: 10000 ms
  VLAN ID: 30
  Priority: 3
  RMEP  MAC             VLAN/PEER       AGE   PORT     SLOTS
  ====  =====           =============== ====  ======== ====
   400  000c.dbe2.8a00      30          879      1/2 1,
   200  000c.dbf5.e500      30         1550      1/2 1,
```

```
RouterB#show cfm connectivity
Domain: CUST_1 Level: 7
  Maintenance association: ma_5
  CCM interval: 10000 ms
  VLAN ID: 30
  Priority: 5
  RMEP   MAC              VLAN/PEER       AGE   PORT     SLOTS
  ====   =====            ========== ====  ======== ====
   400   000c.dbe2.8a00       30          898       1/3 1,
   100   000c.dbe2.b400       30         1569       1/3 1,

RouterC#show cfm connectivity
Domain: CUST_1 Level: 7
  Maintenance association: ma_5
  CCM interval: 10000 ms
  VLAN ID: 30
  Priority: 4
  RMEP   MAC              VLAN/PEER       AGE   PORT     SLOTS
  ====   =====            ========== ====  ======== ====
   200   000c.dbf5.e500       30          907      1/4 1,
   100   000c.dbe2.b400       30          904      1/4 1,
```

Sample configuration for a customer domain using MPLS VLL

The topology inside an MPLS networks can be managed by using LSP ping and LSP trace route to detect and diagnose LSP failures. CFM packets are Ethernet packets with well know CFM etype and are not shown in the MPLS cloud. Therfore, the topology inside MPLS cannot be managed by the CFM protocol. However, you can use CFM to monitor the health of a VPLS or VLL instances.

FIGURE 236 Sample configuration

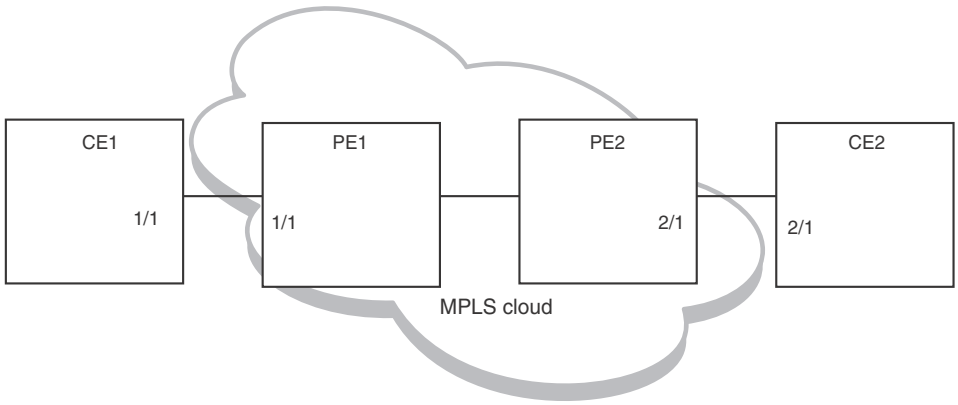


Figure 236 shows a deployment scenario where CE1 and CE2 are customer devices and PE1 and PE2 are provider routers. Port 1/1 on PE1 and port 2/1 on PE2 are VLL-end points. Port 1/1 on PE1 is connected to port 1/1 on CE1 and port 2/1 on PE2 is connected to port 2/1 on CE2.

Achieving end-to-end connectivity between CE1 and CE2

To achieve end-to-end connectivity between CE1 and CE2, configure DOWN MEP on 1/1 and 2/1.PE1 and PE2 acts as MIP. The configuration for this is as follows.

Configuring CE1

1. To enable CFM, enter the following command.

   ```
   CE1(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

   ```
   CE1(config-cfm)#domain-name CUST_1 level 7
   ```

3. Create a maintenance association within a specified domain of vlan-id 30 with a priority 3.

   ```
   CE1(config-cfm-md-CUST_1)#ma-name ma_5 vlan-id 30 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

   ```
   CE1(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
   ```

5. Configure a MEP on port 1/1 and vlan 30.

   ```
   CE1(config-cfm-md-CUST_1-ma-ma_5)#mep 1 down vlan 30 port ethe 1/1
   ```

6. 6. Configure a remote-mep.

   ```
   CE1(config-cfm-md-CUST_1-ma-ma_5)#remote-mep 2 to 2
   ```

Configuring CE2

1. To enable CFM, enter the following command.

   ```
   CE2(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

   ```
   CE2(config-cfm)#domain-name CUST_1 level 7
   ```

3. Create a maintenance association within a specified domain of vlan-id 30 with a priority 3.

   ```
   CE2(config-cfm-md-CUST_1)#ma-name ma_5 vlan-id 30 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

   ```
   CE2(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
   ```

5. Configure a MEP on port 2/1 and vlan 30.

   ```
   CE1(config-cfm-md-CUST_1-ma-ma_5)#mep 2 down vlan 30 port ethe 2/1
   ```

6. Configure a remote-mep.

   ```
   CE1(config-cfm-md-CUST_1-ma-ma_5)#remote-mep 1 to 1
   ```

MPLS Configurations on PE1

Before configuring CFM on PE1, the MPLS Configuration on PE1 must be done.

Enter the following commands to configure VLL peers from PE1 to PE 2.

1. To create a VLL instance, enter commands such as the following.

```
PE1(config)#router mpls
PE1(config-mpls)vll pe1-to-pe2 30
```

2. To specify a VLL peer, enter a command such as the following.

```
PE1(config-mpls-vll)vll-peer 1.1.1.2
```

3. To specify an un-tagged endpoint for a VLL instance, enter the following commands.

```
PE1(config-mpls-vll)untagged ethe 1/1
```
Tagged ports are configured under a VLAN ID.

4. To specify a tagged endpoint for a VLL instance, enter the following commands.

```
PE1(config-mpls-vll)vlan 30
PE1(config-mpls-vll-vlan)tagged ethe 1/1
```

IEEE 802.1ag Configuration on PE1

If the VLL configuration is not done prior to configuring the maintenance association, then the maintenance association is not allowed.

1. To enable CFM, enter the following command.

```
PE1(config)#cfm-enable
```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

```
PE1(config-cfm)#domain-name CUST_1 level 7
```

3. Create a maintenance association within a specified domain of vll-id 30 with a priority 3.

```
PE1(config-cfm-md-CUST_1)#ma-name ma_5 vll-id 30 priority 3
```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

```
PE1(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```
In the above configuration, a MIP (Maintenance Intermediate Point) is created by default on the VLL port. You can also configure an explicit MIP on PE1. In that case, MIP is created on the VLL-port if there is a MEP (Maintenance End Point) created on the port at some lower Maintenance Domain Level.

5. To configure an explicit MIP on PE1, enter the following command.

```
PE1(config-cfm-md-CUST_1-ma-ma_5)# mip-creation explicit
```

6. To change back to default use the following command .

```
PE1(config-cfm-md-CUST_1-ma-ma_5)# mip-creation default
```

MPLS Configurations on PE2

Before configuring CFM on PE2, MPLS is configured on PE2.

Use the following steps to configure VLL peers from PE2 to PE 1.

1. To create a VLL instance, enter commands such as the following.

```
PE2(config)#router mpls
PE2(config-mpls)vll pe2-to-pe1 30
```

2. To specify a VPLS peer enter a command such as the following.

```
PE2(config-mpls-vll)vpls-peer 1.1.1.1
```

3. To specify an un-tagged endpoint for a VLL instance, enter the following commands.

```
PE2(config-mpls-vll)untagged ethe 2/1
```

Tagged ports are configured under a VLAN ID.

4. To specify a tagged endpoint for a VLL instance, enter the following commands.

```
PE2(config-mpls-vll)vlan 30
PE2(config-mpls-vll-vlan)tagged ethe 2/1
```

IEEE 802.1ag Configurations on PE2

If the VLL configuration is not done prior to configuring the maintenance association, then the maintenance association is not allowed.

1. To enable CFM, enter the following command.

```
PE2(config)#cfm-enable
```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

```
PE2(config-cfm)#domain-name CUST_1 level 7
```

3. Create a maintenance association within a specified domain of vll-id 30 with a priority 3 .

```
PE2(config-cfm-md-CUST_1)#ma-name ma_5 vll-id 30 priority 3
```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

```
PE2(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
```

In the above configuration, MIP is created by default on the VLL-endpoint. You can also configure an explicit-mip on PE2. In that case, MIP is created on the VLL-port if there is a MEP is created on the endpoint at some lower MD Level.

5. To configure an explicit-mip on PE2, enter the following command.

```
PE2(config-cfm-md-CUST_1-ma-ma_5)# mip-creation explicit
```

6. To change back to default use the following command.

```
PE2(config-cfm-md-CUST_1-ma-ma_5)# mip-creation default
```

Verifying connectivity using IEEE 802.1ag

Once CE1,CE2,PE1 and PE2 are configured, you can determine the end-to-end connectivity by looking at the remote-mep status by using the following show commands.

```
CE1#show cfm connectivity
Domain: CUST_1 Level: 7
Maintenance association: ma_5
CCM interval: 10000 ms
VLAN ID: 30
Priority: 3
RMEP MAC          VLAN/PEER AGE PORT SLOTS
==== ================== ==== ==== ====
2   000c.dbe2.8a00 30 879 1/2 1,
CE1#show cfm connectivity domain CUST_1 ma ma_5 rmep-id 2
Domain: CUST_1 Level: 7
```

```
Maintenance association: ma_5 VLAN ID: 30 Priority: 3
CCM interval: 10
RMEP  MAC      PORT Oper Age CCM  RDI   Port   Intf  Intvl  Seq
State Val Cnt Status Status  Error  Error
==== =============== === === ===== ===  ===  ====  ====== ====== ====
2      000c.dbe2.8a00 1/1 OK  26000 2600 N     0      0      N      N=
```

Syntax: show cfm connectivity [domain <*name*>**] [ma MA** <*name*>**]**

The [**domain** <*name*>] parameter displays the specific domain information. By default, all defined domains are shown.

The [**ma** <*name*>] parameter specifies the maintenance association name. By default, all defined domains are shown.

TABLE 426 Show CFM connectivity output descriptions

This field...	Displays...
Domain	The Domain is the network or the part of the network for which faults in connectivity are displayed.
Level	The level is the domain level in the range <*0-7*>. The levels can be: • Customer's MD levels: 5 - 7 • Provider's MD levels: 3 - 4 • Operator's MD levels: 0 - 2
Maintenance association	The maintenance association name.
CCM interval	The time interval between two successive Continuity Check messages (CCMs) that are sent by MEPs in the specified Maintenance Domain.
VPLS ID	The VPLS identifier of the maintenance association.
Priority	The priority of the CCM messages, sent by MEPs, in the range <*0-7*>.
RMEP	The remote maintenance end point ID
MAC	DIsplays the associated MAC Address.
VLAN/VC	VLAN ID or VC label learned from the CCM packet. VC label is in hexadecimal format.
Age	Uptime since RMEP discovery or from last age out
PORT	DIsplays the associated port.
SLOTMASK	Mask of slots that are receiving CCM packets which are used for multi-slot trunks. For example a value of 0005 indicates Slots 1 and 3.

Verifying connectivity in a VLL network using IEEE 802.1ag Loopback

You can manually monitor the status of a VLL peer using IEEE 802.1ag CFM Loopback (MAC ping) as shown below.

```
CE1#cfm loopback domain CUST_1 ma ma_5 src-mep 1 target-mep 2
DOT1AG: Sending 10 Loopback to 000c.dbe2.8a00, timeout 10000 msec
Type Control-c to abort
Reply from 000c.dbe2.8a00: time=3ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time=38ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
A total of 10 loopback replies received.
Success rate is 100 percent (10/10), round-trip min/avg/max=0/4/38 ms.
```

Syntax: [no] cfm linktrace domain NAME ma MA-NAME src- mep <*mep-id*> dst-mip HH:HH:HH:HH:HH:HH | target-mep <*mep-id*>} [timeout <*timeout*>] [ttl <*TTL*>]

The **domain** NAME parameter specifies the maintenance domain to be used for a linktrace message. The NAME attribute is case-sensitive.

The **ma** MA NAME parameter specifies the maintenance association to be used for a linktrace message. The NAME attribute is case-sensitive.

The **src-mep** <*mep-id*> parameter specifies the Source ID in the range 1 – 8192.

The **dst-mip** HH:HH:HH:HH:HH:HH parameter specifies the MAC-address of the MIP linktrace destination.

The **dst-mep** <*mepid*> parameter specifies the ID of the linktrace destination.

The **timeout** <*timeout*> parameter specifies the time to wait for a linktrace reply. The range is 1 – 30 seconds.

The **ttl** <*TTL*> parameter specifies the initial TTL field value in the range 1 – 64.The default is 8 seconds

Verifying Connectivity in a VLL Network Using IEEE 802.1ag Linktrace

You can manually monitor the status of a VLL peer using IEEE 802.1ag CFM Loopback (MAC Ping) as shown below.

```
NetIron1#cfm linktrace domain CUST_1 ma ma_5 src-mep 1 target-mep 2
Linktrace to 000c.dbe2.8a00 on Domain CUST_1, level 4: timeout 10ms, 8 hops
-----------------------------------------------------------------
Hops   MAC     Ingress  Ingress Action  Relay Action
Forwarded  Egress   Egress Action    Nexthop
-----------------------------------------------------------------
Type Control-c to abort
1   000c.dbe2.8a00 1.1.1.1   IgrOK          RLY_HIT
Not Forwarded
Destination 000c.dbe2.8a00 reached
```

Syntax: [no] cfm loopback domain NAME ma MA-NAME scr-mep <*mep-id*> {dst-mip HH:HH:HH:HH:HH:HH | target-mep <*mep-id*>} [number <*number*>] [timeout <*timeout*>]

The **domain** NAME parameter specifies the maintenance domain to be used for a linktrace message. The NAME attribute is case-sensitive.

The **ma** MA-NAME parameter specifies the maintenance association to be used for a linktrace message. The MA-NAME attribute is case-sensitive.

The **src-mep** <*mep-id*> parameter specifies the Source ID in the range <*1-8192*>.

The **dst- mip** HH:HH:HH:HH:HH:HH parameter specifies the MAC address of the MIP linktrace destination.

The **dst-mep** <*mep-id*> parameter specifies the Destination ID in the range <*1-8192*>.

The **number** <*number*> parameter specifies the number of loopback messages to be sent.

The **timeout** <*timeout*> parameter specifies the timeout used to wait for linktrace reply.

If the linktrace and loopback to target-mep 2 fails, then the linktrace can be done on the MIPs on PE1 and PE2 to know the exact failure.

Deployment scenario with PEs functioning as DOWN MEP

It is also possible to configure DOWN MEP on VLL end-points. For example, in Figure 236 , the DOWN MEP can be configured to monitor the connectivity between CE1 and PE1 or to monitor the connectivity between CE2 and PE2.

Configuring CE1

1. To enable CFM, enter the following command.

   ```
   CE1(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_2 and level 6.

   ```
   CE1(config-cfm)#domain-name CUST_2 level 6
   ```

3. Create a maintenance association within a specified domain of vlan-id 30 with a priority 3 .

   ```
   CE1(config-cfm-md-CUST_2)#ma-name ma_6 vlan-id 30 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

   ```
   CE1(config-cfm-md-CUST_2-ma-ma_6)#ccm-interval 10-second
   ```

5. Configure a MEP on port 1/1 and vlan 30.

   ```
   CE1 (config-cfm-md-CUST_2-ma-ma_6)#mep 3 down vlan 30 port ethe 1/1
   ```

6. Configure a remote-mep.

   ```
   CE1(config-cfm-md-CUST_1-ma-ma_5)#remote-mep 4 to 4
   ```

Configuring PE1

The MPLS-VLL configuration is the same as shown in the previous deployment scenario. If the VLL configuration is not done prior to configuring maintenance association, the MA configuration will not be allowed. Also the port and vlan in the MEP configuration should exist in the VLL configuration prior to MEP configuration , otherwise it is not allowed. The port in the MEP configuration can be either a tagged or untagged port already present in the VLL configuration.

1. To enable CFM, enter the following command.

   ```
   PE1(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_2 and level 6.

   ```
   PE1(config-cfm)#domain-name CUST_2 level 6
   ```

3. Create a maintenance association within a specified domain of vll-id 30 with a priority 3 .

```
PE1(config-cfm-md-CUST_2)#ma-name ma_6 vll-id 30 priority 3
```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

```
PE1(config-cfm-md-CUST_2-ma-ma_6)#ccm-interval 10-second
```

5. Configure a MEP on port 1/1 and vlan 30

```
CE-1 (config-cfm-md-CUST_2-ma-ma_6)#mep 4 down vlan 30 port ethe 1/1
```

To monitor the connectivity between CE-1 and PE-1, you can use the **show cfm connectivity** commands as mentioned in the previous scenario. You can also use the **loopback** or **linktrace** commands on CE-1 or PE-1.

Deployment scenario with PEs functioning as UP MEP

UP MEPs can also be configured on PEs. This monitors connectivity of VLL end points.

Configuring PE1

The MPLS-VLL configuration is the same as shown in the previous deployment scenario. If the VLL configuration is not done prior to configuring maintenance association,the MA configuration would not be allowed. Also the port and vlan in the MEP configuration should exist in VLL configuration prior to MEP configuration , otherwise it will not be allowed. The port in the MEP configuration can be either a tagged or untagged port already present in VLL configuration.

1. To enable CFM, enter the following command.

```
PE1(config)#cfm-enable
```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

```
PE1(config-cfm)#domain-name PROVIDER_1 level 4
```

3. Create a maintenance association within a specified domain of vll-id 30 with a priority 3 .

```
PE1(config-cfm-md-PROVIDER_1)#ma-name ma_8 vll-id 30 priority 3
```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

```
PE1(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
CE1 (config-cfm-md-PROVIDER_1-ma-ma_8)#mep 6 up vlan 30 port ethe 1/1
```

Configuring PE2

The configuration on PE1 is similar to the PE1 configuration.

1. To enable CFM, enter the following command.

```
PE2(config)#cfm-enable
```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

```
PE2(config-cfm)#domain-name PROVIDER_1 level 4
```

3. Create a maintenance association within a specified domain of vll-id 30 with a priority 3 .

```
PE1(config-cfm-md-PROVIDER_1)#ma-name ma_8 vll-id 30 priority 3
```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

```
PE1(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
CE1 (config-cfm-md-PROVIDER_1-ma-ma_8)#mep 7 up vlan 30 port ethe 2/1
```

To monitor the connectivity between PE1 and PE2, you could use the "`show cfm connectivity`" commands as mentioned in the previous scenario. Also you could use either loopback or linktrace on PE1 or PE2.

Configuring PE2

1. To enable CFM, enter the following command.

```
PE2(config)#cfm-enable
```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

```
PE2config-cfm)#domain-name PROVIDER_1 level 4
```

3. Create a maintenance association within a specified domain of vll-id 30 with a priority 3 .

```
PE1(config-cfm-md-PROVIDER_1)#ma-name ma_8 vll-id 30 priority 3
```

4. Set the time interval between successive Continuity Check Messages.The default is 10-seconds.

```
PE1(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
```

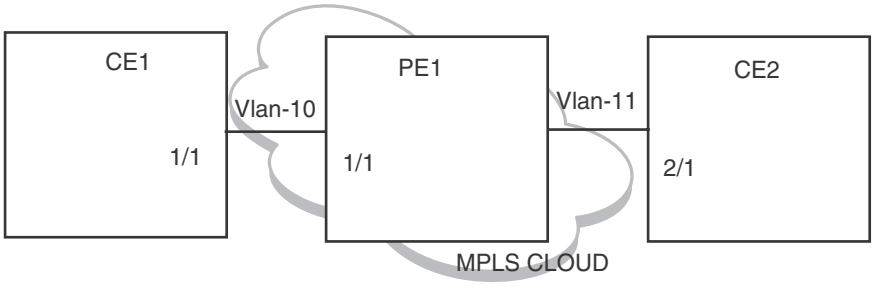
5. Configure MEP 4 down on port 1/1 and vlan 30

```
CE-1 (config-cfm-md-CUST_2-ma-ma_6)#mep 4 down vlan 30 port ethe 1/1
```

To monitor the connectivity between PE-1 and PE-2, we could use "**show cfm connectivity**" commands as mentioned in the previous scenario. Also we could use either loopback or linktrace on PE-1 or PE-2.

IEEE 802.1ag with VLL-LOCAL

FIGURE 237 IEEE 802.1ag over VLL-LOCAL



In the case of IEEE 802.1ag over VLL-LOCAL, the PE acts as a MIP and VLL does VLAN translation. As shown in Figure 237. MEP is configured on vlan-10 on CE1 and vlan-11 on CE2. On PE1 , MIP is configured on VLL-LOCAL and which has vlan-10, port 1/1 and vlan-11,port 2/1 configured as end points.

UP MEP would not be allowed for VLL-Local.

MPLS configurations on PE1

Before configuring CFM on PE1 we need to do MPLS Configuration on PE1.

Enter the following commands to configure VLL peers from PE1 to PE 2.

1. To create a VLL instance, enter commands such as the following.

```
PE1(config)#router mpls
PE1(config-mpls)vll-local test1
```

2. To specify an un-tagged endpoint for a VLL instance, enter the following commands.

```
PE1(config-mpls-vll-test1)untagged ethe 1/1
```

Tagged ports are configured under a VLAN ID.

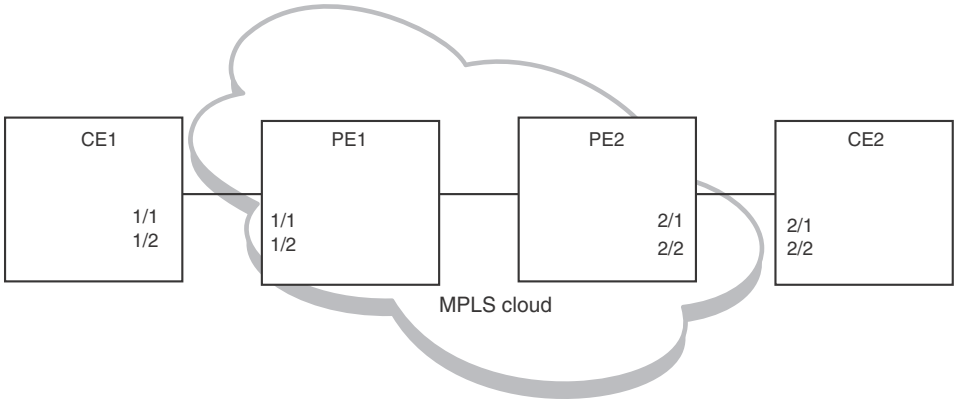
3. To specify a tagged endpoint for a VLL instance, enter the following commands.

```
PE1(config-mpls-vll-test1)vlan 30
PE1(config-mpls-vll-vlan)tagged ethe 1/1
```

As in the previous case, to monitor the connectivity between CE1 and CE2, you can use "**show cfm connectivity**" commands as mentioned in the previous scenario. Also we could use either loopback or linktrace on CE1 or CE2.

LAG-support for IEEE 802.1ag-over-vll

FIGURE 238 - IEEE 802.1ag over VLL scenerio



As shown in Figure 238, you can have MEP configuration over a LAG port. On CE1 and CE2 DOWN MEP is configured on VLAN and on PE1 and PE2 DOWN or UP MEP would be configured, depending on what to monitor.

The configuration and monitoring of MEPs is similar as mentioned in the previous examples.

Deletion of VLL

NOTE
Deletion of VLL would cause the deletion of Maintenance Association and corresponding MEPs on that MA.

Sub-second timer support

The ccm-interval command sets the time interval between two successive Continuity Check messages (CCMs) that are sent by MEPs in the specified Maintenance Domain. The default value is 10 seconds. There is support for sub-second timers 3.3 ms, 10 ms and 100 ms. As in the case of VLAN and VPLS, for sub-second timers pbif hardware assist is used to transmit and process the CCM packets.

NOTE
The sub-second timer functionality is not supported on VLL-Local.

Hitless upgrade support

Hitless upgrade support for IEEE 802.1ag over VLL is similar to IEEE 802.1ag hitless upgrade support for VLAN/VPLS.

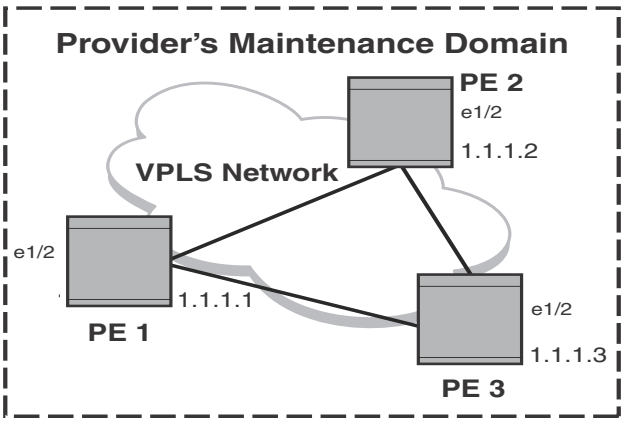
Monitoring the status of devices in a VPLS network in a Provider's Maintenance Domain

CFM provides capabilities to detect, verify, and isolate connectivity failures.

NOTE
When configuring 802.1ag over VPLS, if the VPLS endpoint is deleted from the configuration, the MEP configuration is deleted under CFM without warning.

In the Figure 239, CFM is applied over a VPLS network; ports 1/2 and 1/3 are customer facing networks; and port 1/1 is an uplink to a VPLS cloud.

FIGURE 239 VPLS cloud with CFM enabled



Configuring PE 1

1. To enable CFM for VPLS, enter the following command.

```
PE1(config)#cfm-enable
```

2. Create a maintenance domain with a specified name **VPLS-SP** and level **4**.

```
PE1(config-cfm)#domain-name VPLS-SP level 4
```

3. Create a maintenance association within a specified domain of **vpls-id 1** with a priority **3** .

```
PE1(config-cfm-md-VPLS-SP)#ma-name ma_1 vpls-id 1 priority 3
```

4. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port **1/2** as MEP to a specified maintenance association.

```
PE1(config-cfm-md-VPLS-SP-ma-ma_1)#mep 1 up vlan 30 port ethe 1/2
```

MPLS configurations

Enter the following commands to configure VPLS peers from PE 2 to PE3.

1. To create a VPLS instance, enter commands such as the following.

```
PE1(config)#router mpls
PE1(config-mpls)#vpls 1 1
```

2. To specify two remote VPLS peers within a VPLS instance, enter a commands such as the following.

```
PE1(config-mpls-vpls-1)#vpls-peer 1.1.1.2
PE1(config-mpls-vpls-1)#vpls-peer 1.1.1.3
```

3. Tagged ports are configured under a VLAN ID. To specify a tagged endpoint for a VPLS instance, enter the following commands.

```
PE1(config-mpls-vpls-1)#vlan 30
PE1(config-mpls-vpls-1-vlan-30)#tagged ethe 1/2 to 1/3
```

Configuring PE 2

CFM configuration steps for Router 2 are listed below.

1. To enable CFM, enter the following command.

```
PE2(config)#cfm-enable
```

2. Create a maintenance domain with a specified name **VPLS-SP** and level **4**.

```
PE2(config-cfm)#domain-name VPLS-SP level 4
```

3. Create a maintenance association within a specified domain of **vpls-id 1** with a priority **3** .

```
PE2(config-cfm-VPLS-SP)#ma-name ma_1 vpls-id 1 priority 3
```

4. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port **1/2** as MEP to a specified maintenance association.

```
PE2(config-cfm-md-VPLS-SP-ma-ma_1)#mep 2 up vlan 30 port ethe 1/2
```

MPLS configurations

Enter the following commands to configure VPLS peers from PE1 toPE 3.

1. To create a VPLS instance, enter commands such as the following.

```
PE2(config)#router mpls
PE2(config-mpls)vpls 1 1
```

2. To specify two remote VPLS peers within a VPLS instance, enter a command such as the following.

```
PE2(config-mpls-vpls-1)vpls-peer 1.1.1.1
PE2(config-mpls-vpls-1)vpls-peer 1.1.1.3
```

Tagged ports are configured under a VLAN ID. To specify a tagged endpoint for a VPLS instance, enter the following commands.

```
PE2(config-mpls-vpls-1)vlan 30
PE2(config-mpls-vpls-1-vlan-30)tagged ethe 1/2
```

Configuring PE 3

CFM configuration steps for PE 3 are listed below.

1. To enable CFM for VPLS, enter the following command.

```
PE3(config)#cfm-enable
```

2. Create a maintenance domain with a specified name **VPLS-SP** and level **4**.

```
PE3(config-cfm)#domain-name VPLS-SP level 4
```

3. Create a maintenance association within a specified domain of **vpls-id 1** with a priority **3** .

```
PE3(config-cfm-md-VPLS-SP)#ma-name ma_1 vpls-id 1 priority 3
```

4. Configuring a MED for each of the Domain Service Access Points of a service instance creates a MA to monitor the connectivity. Add ethernet port 1/2 as MEP to a specified maintenance association.

```
PE3(config-cfm-md-VPLS-SP-ma-ma_1)#mep 3 up vlan 30 port ethe 1/2
```

MPLS configurations

Enter the following commands to configure VPLS peers from Router 1 to Router 2.

1. To create a VPLS instance, enter commands such as the following.

```
PE3(config)router mpls
PE3(cconfig-mpls)vpls 1 1
```

2. To specify two remote VPLS peers within a VPLS instance, enter a command such as the following.

```
PE3(config-mpls-vpls-1)vpls-peer 1.1.1.1
PE3(config-mpls-vpls-1)vpls-peer 1.1.1.2
```

3. Tagged ports are configured under a VLAN ID. To specify a tagged endpoint for a VPLS instance, enter the following commands.

```
PE3(config-mpls-vpls-1)vlan 30
PE3(config-mpls-vpls-1-vlan-30)tagged ethe 1/2
```

Verifying connectivity in a VPLS network using IEEE 802.1ag

To display VPLS IEEE 802.1ag connectivity, enter the following commands.

```
NetIron#sh cfm domain VPLS-SP
Domain: VPLS-SP
Level: 4
  Maintenance association: ma_1
  CCM interval: 10
  VPLS ID: 1
  Priority: 3
  MEP   Direction  MAC             PORT
  ====  =========  =========       ====
     1    UP       000c.dbe3.8210  ethe 1/3
```

Syntax: show cfm [domain NAME] [ma NAME]

The **domain NAME** parameter displays the specific domain information. By default, all defined domains are shown.

The [**ma NAME**] parameter specifies the maintenance association name. By default, all defined domains are shown.

TABLE 427 Output for show CFM domain command

This field...	Displays...
Domain	The Domain is the network or the part of the network for which faults in connectivity are displayed.
Level (Maintenance Domain)	The level is the domain level in the range <*0-7*>. The levels can be: • Operator's MD levels: 0 - 2 • Provider's MD levels: 3 - 4 • Customer's MD levels: 5 – 7
Maintenance association	The maintenance association name.
CCM interval	The time interval between two successive Continuity Check messages (CCMs) that are sent by MEPs in the specified Maintenance Domain.
VLAN ID	The VLAN identifier of the maintenance association.
VPLS ID	The VPLS identifier of the maintenance association.
Priority	The priority of the CCM messages, sent by 'MEPs, in the range <*0-7*>.
MEP	The maintenance end point ID
Direction	Displays the direction the MEP was sent: Up - The MEP direction away from the monitored VLAN. Down - The MEP direction is towards the monitored VLAN.
MAC	Displays the associated MAC Address.
PORT	Displays the associated port.

The **show cfm connectivity** command, displays connectivity statistics for the remote database. For the **show cfm connectivity** command to take effect, CFM should first be enabled in the Protocol Configuration mode.

```
NetIron#show cfm connectivity
  Domain: VPLS-SP Level: 4
  Maintenance association: ma_1
  CCM interval: 10
  VPLS ID: 1
  Priority: 3
  RMEP     MAC          VLAN/VC AGE   PORT SLOTMASK
  ====    =====         ======= ====  ==== ====
     4  000c.dbe2.d80a  00f00a1 2157       0008
     2  000c.dbe2.b560  00f00a0 2597       0008


NetIron#show cfm connectivity domain VPLS-SP ma ma_1 rmep-id 2
Domain: VPLS-SP Level: 4
Maintenance association: ma_1 VPLS ID: 1 Priority: 3
CCM interval: 10
  RMEP   MAC            PORT    Oper  Age  CCM  RDI Port   Intf   Intvl Seq
                                 State Val  Cnt      Status Status Error Error
  ====  =============== ======= ===== ===  === ==== ====== ====== ===== =====
     2  000c.dbe2.b560  00f00a0   OK  26000 2600   N     0      0     N     N
```

Syntax: show cfm connectivity[domain NAME] [ma MA NAME]

The [**domain NAME**] parameter displays information for a specific domain. By default, all defined domains are shown.

The [**ma NAME**] parameter specifies the maintenance association name. By default, all defined domains are shown.

TABLE 428 Output for show CFM connectivity command

This field...	Displays...
Domain	The Domain is the network or the part of the network for which faults in connectivity are displayed.
Level	The level is the domain level in the range <*0-7*>. The levels can be: • Customer's MD levels: 0 - 2 • Provider's MD levels: 3 - 4 • Operator's MD levels: 5 – 7
Maintenance association	The maintenance association name.
CCM interval	The time interval between two successive Continuity Check messages (CCMs) that are sent by MEPs in the specified Maintenance Domain.
VLAN ID	The VLAN identifier of the maintenance association.
VPLS ID	The VPLS identifier of the maintenance association.
Priority	The priority of the CCM messages, sent by 'MEPs, in the range <*0-7*>.
RMEP	The remote maintenance end point ID
MAC	DIsplays the associated MAC Address.
PORT	DIsplays the associated port.
Oper State	Defines the state of the port attached to the MEP. Possible values OK and Fail
Age Val	Age of the operational state of the port.

TABLE 428 Output for show CFM connectivity command

This field...	Displays...
CCM Count	Displays the total number of Continuity Check messages (CCMs) that are sent.
RDI	Remote Defect Indicator
Port Status	The status of a port
Intf Status	The status of the interface
Intvl Error	Displays Y if there has been an interval error and N if no interval errors have been detected.
Seq Error	Displays Y if there has been a sequence error and N if no sequence errors have been detected.

Verifying connectivity in a VPLS network using IEEE 802.1ag Loopback

You can manually monitor the status of VPLS peers using IEEE 802.1ag CFM Linktrace (MAC traceroute) and CFM Loopback (MAC Ping) as shown below.

```
PowerConnect#cfm linktrace domain VPLS-SP ma ma_1 src-mep 1 target-mep 4
Linktrace to 000c.dbe2.d80a on Domain VPLS-SP, level 4: timeout 10ms, 8 hops
----------------------------------------------------------------
Hops           MAC             Ingress     Ingress Action   Relay Action
            Forwarded          Egress      Egress Action    Nexthop
----------------------------------------------------------------

Type Control-c to abort
   1    000c.dbe2.d80a               1.1.1.1      IgrOK      RLY_HIT
        Not Forwarded
Destination 000c.dbe2.d80a reached
```

Syntax: [no] cfm linktrace domain NAME ma MA-NAME src- mep <*mep-id*> dst-mip HH:HH:HH:HH:HH:HH | target-mep <*mep-id*>} [timeout <*timeout*>] [ttl <*TTL*>]

The **domain NAME** parameter specifies the maintenance domain to be used for a linktrace message. The NAME attribute is case-sensitive.

The **ma MA NAME** parameter specifies the maintenance association to be used for a linktrace message. The NAME attribute is case-sensitive.

The **src-mep** <*mep-id*> parameter specifies the Source ID in the range <*1-8192*>.

The **dst-mip** HH:HH:HH:HH:HH:HH parameter specifies the MAC-address of the MIP linktrace destination.

The **dst-mep** <*mepid*> parameter specifies the ID of the linktrace destination.

The **timeout** <*timeout*> parameter specifies the timeout used to wait for linktrace reply. The default value is < *1-30* > seconds.

The **ttl** <*TTL*> parameter specifies the initial TTL field value in the range < *1-64*>.The default is 8 seconds.

```
PowerConnect#cfm loopback domain VPLS-SP ma ma_1 src-mep 1 target-mep 4
DOT1AG: Sending 10  Loopback to 000c.dbe2.d80a, timeout 10000 msec
Type Control-c to abort
Reply from 000c.dbe2.d80a: time=3ms
Reply from 000c.dbe2.d80a: time<1ms
Reply from 000c.dbe2.d80a: time<1ms
Reply from 000c.dbe2.d80a: time<1ms
Reply from 000c.dbe2.d80a: time=38ms
Reply from 000c.dbe2.d80a: time<1ms
Reply from 000c.dbe2.d80a: time<1ms
Reply from 000c.dbe2.d80a: time<1ms
Reply from 000c.dbe2.d80a: time<1ms
Reply from 000c.dbe2.d80a: time<1ms
A total of 10 loopback replies received.
Success rate is 100 percent (10/10), round-trip min/avg/max=0/4/38 ms.
#
```

Syntax: [no] cfm loopback domain NAME ma MA-NAME scr-mep <*mep-id*> {dst-mip HH:HH:HH:HH:HH:HH | target-mep <*mep-id*>} [number <*number*>] [timeout <*timeout*>]

The **domain** NAME parameter specifies the maintenance domain to be used for a linktrace message. The NAME attribute is case-sensitive.

The **ma** MA-NAME parameter specifies the maintenance association to be used for a linktrace message. The MA-NAME attribute is case-sensitive.

The **src-mep** <*mep-id*> parameter specifies the Source ID in the range <*1-8192*>.

The **dst- mip** HH:HH:HH:HH:HH:HH parameter specifies the MAC address of the MIP linktrace destination.

The **dst-mep** <*mep-id*> parameter specifies the Destination ID in the range <*1-8192*>.

The **number** <*number*> parameter specifies the number of loopback messages to be sent.

The **timeout** <*timeout*> parameter specifies the timeout used to wait for linktrace reply.

You have to configure MAs with different MD levels to monitor the different endpoints with different

NOTE
You have to configure MAs with different MD levels to monitor the different endpoints with different VLAN IDs in the same VPLS instance.

Syslog message

If CFM is configured, a syslog message will be generated when remote MEPs change their states or if there are service cross connections.

Sample Syslog Messages

```
NetIron#
SYSLOG: Jan  7 11:22:55:<9>Router2, DOT1AG: Remote MEP 4 in Domain VPLS-SP, MA
ma_1 aged out
SYSLOG: Jan  7 11:23:13:<9>Router2, DOT1AG: Remote MEP 4 in Domain VPLS-SP, MA
ma_1 become UP state
```

When a failure is detected within a VPLS cloud, use LSP Ping and Traceroute. Refer to "LSP ping and traceroute" on page 2229 for additional information.

Support for IEEE 802.1ag CFM for Provider Bridges (PB) and Provider Backbone Bridges (PBB)

The PowerConnect support the following single tagging and double tagging cases:

- MEP (up/down) and MIP on C-VLANs
- MEP (up/down) and MIP on S-VLANs - The ability to change tag-type 88a8 to S-VLANs

IEEE 802.3ah EFM-OAM

The IEEE 802.3ah Ethernet in the First Mile (EFM) is supported on the PowerConnect devices.

The IEEE 802.3ah Ethernet in the First Mile (EFM) standard specifies the protocols and Ethernet interfaces for using Ethernet over access links as a first-mile technology and transforming it into a highly reliable technology.

Using the Ethernet in the First Mile solution, the user will gain broadcast Internet access, in addition to services, such as Layer 2 transparent LAN services, Voice services over Ethernet Access networks, and Video and multicast applications, reinforced by security and Quality of Service control in order to build a scalable network.

The in-band management specified by this standard defines the operations, administration and maintenance (OAM) mechanism needed for the advanced monitoring and maintenance of Ethernet links in the first mile. The OAM capabilities facilitate network operation and troubleshooting. Basic 802.3 frames convey OAM data between two ends of the physical link. EFM OAM is optional and can be disabled on each physical port.

OAM initiatives are classified into three layers: transport, connectivity and service. The transport layer is the collection of forwarding entities and interconnected segments that form a multi-hop Ethernet network, and provide connectivity between devices. The transport layer OAM is specified by the IEEE 802.3ah (Clause 57) and provides single-link OAM capabilities. When OAM is present, two connected OAM sub-layers exchange protocol data units (OAMPDUs). OAM PDUs are standard-size frames that can be sent at a maximum rate of 10 frames per second. This limitation is necessary for reducing the impact on the usable bandwidth. It is possible to send each frame several times in order to increase the probability of reception. A combination of the destination MAC address, the Ethernet type/length field and Subtype allow distinguishing OAM PDU frames from other frames.

OAM functionality is designed to provide reliable service assurance mechanisms for both provider and customer networks.

The IEEE 802.3ah EFM standard offers an opportunity to create the operations, OAM sub-layer within the data-link layer of the OSI protocol stack. The data-link layer provides utilities for monitoring and troubleshooting Ethernet links.

Possible applications

The data-link layer OAM is targeted at last-mile applications and service providers can use it for demarcation point OAM services.

Ethernet Last Mile applications require robust infrastructure that is both passive and active. 802.3ah OAM aims to solve validation and testing problems in such an infrastructure.

Using the Ethernet demarcation service providers can additionally manage the remote device without utilizing an IP layer. This can be done by using link-layer SNMP counters, request and reply, loopback testing and other techniques.

EFM- OAM protocol

The functionality of the OAM-EFM can be summarized under the following categories:

Discovery: Discovery is the mechanism to detect the presence of an OAM sublayer on the remote device. During the discovery process, information about OAM entities, capabilities and configuration are exchanged

Link monitoring: This process is used to detect link faults and to provide information about the number of frame errors and coding symbol errors.

Remote fault detection: Provides a mechanism for an OAM entity to convey error conditions to its peer via a flag in the OAMPDUs.

Remote loopback: This mechanism is used to troubleshoot networks and to isolate problem segments in a large network by sending test segments.

Discovery

This is the first phase of the EFM-OAM. At this phase, EFM-OAM identifies network devices along with their OAM capabilities. The Discovery process relies on the Information OAMPDUs (discussed below). During discovery, the following information is advertised through the TLVs within periodic Information OAMPDUs:

- OAM configuration (capabilities): Advertises the capabilities of the local OAM entity. Using this information, a peer can determine what functions are supported and accessible (e.g. loopback capability)
- OAM mode: This is conveyed to the remote OAM entity. The mode can be either active or passive, and can also be used to determine device's functionality
- OAMPDU configuration: This includes maximum OAMPDU size to delivery. In combination with the limited rate of ten frames/sec this information can be used to limit the bandwidth allocated to OAM traffic.

Timers

- Two configurable timers control the protocol, one determining the rate at which OAMPDUs are to be sent, and the second controlling the rate at which OAMPDUs are to be received to maintain the adjacency between devices.
- An additional 1-second non-configurable timer is used for error aggregation, which is necessary for the Link Monitoring Process to generate link quality events.
- The timer should generate PDUs in the range of 1s - 10sec. The default value is 1sec.
- The Hold timer should assume the peer is dead if no packet is received for a period of 1s to 10s. The default value is 5 seconds.

Flags

- Included in every OAMPDU is a flags field, which contains, besides other information, the status of the discovery process. There are three possible values for the status:
- Discovering: Discovery is in progress.

- Stable: Discovery is completed. Once aware of this, the remote OAM entity can start sending any type of OAMPDU.
- Unsatisfied: When there are mismatches in the OAM configuration that prevent OAM from completing the discovery, the discovery process is considered unsatisfactory and cannot continue.

Process overview

The discovery process allows a local Data Terminating Entity (DTE) to detect OAM on a remote DTE. Once OAM support is detected, both ends of the link exchange state and configuration information (such as mode, PDU size, loopback support, etc.). If both DTEs are satisfied with the settings, OAM is enabled on the link. However, the loss of a link or a failure to receive OAMPDUs for five seconds may cause the discovery process the start over again.

DTEs may either be in active or passive mode. Active mode DTEs instigate OAM communications and can issue queries and commands to a remote device. Passive mode DTEs generally wait for the peer device to instigate OAM communications and respond to, but do not instigate, commands and queries. Rules of what DTEs in active or passive mode can do are discussed in the following sections.

Rules for active mode

A DTE in Active mode:

- Initiates the OAM Discovery process
- Sends Information PDUs
- May send Event Notification PDUs
- May send Variable Request/Response PDUs
- May send Loopback Control PDUs.

Exceptions:

- Does not respond to Variable Request PDUs from DTEs in Passive mode
- Does not react to Loopback Control PDUs from DTEs in Passive mode.

Rules for passive mode

A DTE in Passive mode:

- Waits for the remote device to initiate the Discovery process
- Sends Information PDUs
- May send Event Notification PDUs
- May respond to Variable Request PDUs
- May react to received Loopback Control PDUs
- Is not permitted to send Variable Request or Loopback Control OAMPDUs

Link monitoring process

The Link Monitoring Process is used for detecting and indicating link faults under a variety of circumstances. Link monitoring uses the Event Notification OAMPDU, and sends events to the remote OAM entity when there are problems detected on the link. The error events defined in the standard are:

- Errored Symbol Period (errored symbols per second): the number of symbol errors that occurred during a specified period exceeded a threshold. These are coding symbol errors (for example, a violation of 4B/5B coding)
- Errored Frame (errored frames per second): the number of frame errors detected during a specified period exceeded a threshold
- Errored Frame Period (errored frames per N frames): the number of frame errors within the last N frames has exceeded a threshold
- Errored Frame Seconds Summary (errored secs per M seconds): the number of errored seconds (one second intervals with at least one frame error) among the last M seconds has exceeded a threshold.

Since 802.3ah OAM does not guarantee the delivery of OAMPDUs, the Event Notification OAMPDU (discussed in the OAMPDU section below) can be sent multiple times to reduce the probability of losing notifications. A sequence number is used to recognize duplicate events. The Link Monitoring Process operates for all the links on which EFM OAM is enabled

Remote failure indication

Faults in Ethernet that are caused by slowly deteriorating quality are more difficult to detect than completely disconnected links. A flag in the OAMPDU allows an OAM entity to send failure conditions to its peer. The failure conditions are defined as follows:

- Link Fault: The Link Fault condition is detected when the receiver loses the signal. This condition is sent once per second in the Information OAMPDU.
- Dying Gasp: This condition is detected when the receiver goes down. The Dying Gasp condition is considered as unrecoverable. Conditions for dying gasp:
  - Reload or reset from MP
  - Interface disable (admin shutdown)
  - Link-OAM disable on interface (deconfiguration)
  - Crash on the box
- Device power down (incidental or deliberate).
- Critical Event: When a critical event occurs, the device is unavailable as a result of malfunction, and it is to be restarted by the user. The critical events can be sent immediately and continually.

When the dying gasp or critical event occurs, the device driver will call a special API in the EFMOAM implementation.

The link fault applies only when the physical sublayer is capable of independent transmission and reception.

When a link receives no signal from its peer at the physical layer (for example, if the peer's laser is malfunctioning), the local entity sets this flag to let the peer know that its transmit path is inoperable. The link-down API will be called by the device driver in order to notify the remote device of the link fault.

Since the above conditions are severe, when they are set in the flag, the OAMPDU is not subject to normal rate limiting policy.

Remote loopback

An OAM entity can put its remote entity into loopback mode using a loopback control OAMPDU. This helps users ensure quality of links during installation or when troubleshooting. In loopback mode, each frame received is transmitted back on that same port except for OAMPDUs and pause frames. The periodic exchange of OAMPDUs must continue while in the loopback state to maintain the OAM session. The loopback command is acknowledged by responding with an Information OAMPDU with the loopback state indicated in the state field. This allows to estimate if a network segment can satisfy an SLA.

Enabling and disabling EFM-OAM

The link-oam command, in Protocol Configuration mode, enables and disables the EFM-OAM protocol and enters into the EFM-OAM Protocol Configuration mode. The link-oam disable and enable command resets all link-oam parameters to default values.

By default, EFM-OAM is disabled.

To enable EFM-OAM, enter a command such as the following:

```
PowerConnect(config)link-oam
PowerConnect(config-link-oam)#enable
```

Syntax: [no]link-oam

Syntax: enable

The **no** form of the command sets the 802.3ah EFM-OAM to the disabled state.

Specifying the timeout value

The timeout command is a hold down timer that specifies the number of seconds before it declares that the other side has stopped sending OAMPDUs.

```
PowerConnect(config-link-oam) #timeout 10
```

Syntax: [no] timeout <value>

The **no** form of the command restores the default value of 5 OAMPDUs.

The <**value**> parameter specifies the number of seconds before declaring the remote as down. in the range of <1-10>.

Specifying the PDU rate

To set the number of PDUs to be transmitted per second, use the pdu-rate command. The default value is 1.

```
PowerConnect(config-link-oam) #pdu-rate 10
```

Syntax: [no] pdu-rate <value>

The <**value**> parameter specifies the number of PDUs in the range of <1-10>.

The **no** form of the command restores the default value of 1.

Enabling and disabling the EFM-OAM state on the specified interface

The ethernet <slot/port> link-oam command, in Interface Configuration mode, enables and disables EFM-OAM on the specified interface and sets its mode to active or passive.

When both peers are in passive mode (abnormal configuration), the information from “Remote Status” is not updated anymore and it may be inaccurate. By default port state is disabled.

For the link-oam command to take effect, EFM-OAM should first be enabled in the Protocol Configuration mode.

```
PowerConnect(config-link-oam)# ethernet 2/1 link-oam active
```

Syntax: [no]ethernet <slot/port> **link-oam** {active | passive}

When active mode is specified, the device can send OAMPDU packets over this port in order to initiate an EFM-OAM discovery process. For the discovery process to be initiated the EFM-OAM protocol must have been enabled.

When passive mode is specified, the device cannot use this port to send OAMPDU packets, but can respond if it receives OAMPDUs from remote.

The **no** form of the command sets the 802.3ah EFM-OAM to the disabled state.

Display information

The following show commands will display OAM information.

DIsplaying OAM information

To show OAM information on all OAM enabled ports, enter a command such as the following:

```
PowerConnect#show link-oam info
Ethernet Link Status     OAM Status      Mode       Local Stable    Remote Stable
1/1      up              up              active     satisfied       satisfied
1/2      up              up              passive    satisfied       satisfied
1/3      up              up              active     satisfied       satisfied
1/4      up              init            passive    unsatisfied     unsatisfied
1/5      down            down            passive    unsatisfied     unsatisfied
1/6      down            down            passive    unsatisfied     unsatisfied
1/7      down            down            passive    unsatisfied     unsatisfied
```

Displaying detailed information from a specific port

To show detailed OAM information, enter a command such as the following:

```
PowerConnect#show link-oam info detail
OAM information for Ethernet port: 1/1
        +link-oam mode:        active
        +link status:          up
        +oam status:           up
        Local information
                multiplexer action:  forward
                parse action:        forward
                stable:              satisfied
                state:               up
                loopback state:      disabled
                dying-gasp:          false
                critical-event:      false
                link-fault:          false
        Remote information
                multiplexer action:  forward
                parse action:        forward
                stable:              satisfied
                loopback support:    disabled
                dying-gasp:          false
                critical-event:      false
                link-fault:          false
OAM information for Ethernet port: 1/2
        +link-oam mode:        passive
        +link status:          up
        +oam status:           up
        Local information
                multiplexer action:  forward
                parse action:        forward
                stable:              satisfied
                state:               up
                loopback state:      disabled
                dying-gasp:          false
                critical-event:      false
                link-fault:          false
        Remote information
                multiplexer action:  forward
                parse action:        forward
                stable:              satisfied
                loopback support:    disabled
                dying-gasp:          false
                critical-event:      false
                link-fault:          false
```

Syntax: show link-oam info detail ethernet < all I slot/port>

To show detailed OAM information on a specific ethernet port, enter a command such as the following:

```
PowerConnect#show link-oam info detail ethernet 1/1
OAM information for Ethernet port: 1/1
        +link-oam mode:        active
        +link status:          up
        +oam status:           up
        Local information
                multiplexer action:  forward
                parse action:        forward
                stable:              satisfied
                state:               up
                loopback state:      disabled
                dying-gasp:          false
                critical-event:      false
                link-fault:          false
        Remote information
                multiplexer action:  forward
                parse action:        forward
                stable:              satisfied
                loopback support:    disabled
                dying-gasp:          false
                critical-event:      false
                link-fault:          false
```

Syntax: show link-oam info detail [all | ethernet <slot/port>]

DIsplaying OAM statistics

To show OAM statistics, enter a command such as the following:

```
PowerConnect#show link-oam statistics
Ethernet Tx Pdus         Rx Pdus
1/1      507             362
1/2      358             359
1/3      480             355
1/4      0               0
1/5      0               0
1/6      0               0
1/7      0               0
1/8      0               0
1/9      0               0
1/10     0               0
```

Syntax: show link-oam statistics

DIsplaying detailed OAM statistics

To show detailed OAM statistics, enter a command such as the following:

```
PowerConnect#show link-oam statistics detail
OAM statistics for Ethernet port: 1/1
        Tx statistics
                information OAMPDUs:                   587
                loopback control OAMPDUs:              0
                variable request OAMPDUs:              0
                variable response OAMPDUs:             0
                unique event notification OAMPDUs:     0
                duplicate event notification OAMPDUs:  0
                oranization specific OAMPDUs:          0
```

```
                link-fault records:                   0
                critical-event records:               0
                dying-gasp records:                   0
        Rx statistics
                information OAMPDUs:                  442
                loopback control OAMPDUs:             0
                variable request OAMPDUs:             0
                variable response OAMPDUs:            0
                unique event notification OAMPDUs:    0
                duplicate event notification OAMPDUs: 0
                oranization specific OAMPDUs:         0
                unsupported OAMPDUs:                  0
                link-fault records:                   0
                critical-event records:               0
                dying-gasp records:                   0
                discarded TLVs:                       0
                unrecognized TLVs:                    0
OAM statistics for Ethernet port: 1/2
        Tx statistics
                information OAMPDUs:                  440
                loopback control OAMPDUs:             0
                variable request OAMPDUs:             0
                variable response OAMPDUs:            0
                unique event notification OAMPDUs:    0
                duplicate event notification OAMPDUs: 0
                oranization specific OAMPDUs:         0
                link-fault records:                   0
                critical-event records:               0
                dying-gasp records:                   0
        Rx statistics
                information OAMPDUs:                  441
                loopback control OAMPDUs:             0
                variable request OAMPDUs:             0
                variable response OAMPDUs:            0
                unique event notification OAMPDUs:    0
                duplicate event notification OAMPDUs: 0
                oranization specific OAMPDUs:         0
                unsupported OAMPDUs:                  0
                link-fault records:                   0
                critical-event records:               0
                dying-gasp records:                   0
                discarded TLVs:                       0
                unrecognized TLVs:                    0
```

To show detailed OAM statistics, enter a command such as the following:

```
PowerConnect#show link-oam statistics detail ports ethernet 1/1
OAM statistics for Ethernet port: 1/1
        Tx statistics
                information OAMPDUs:                    827
                loopback control OAMPDUs:               0
                variable request OAMPDUs:               0
                variable response OAMPDUs:              0
                unique event notification OAMPDUs:      0
                duplicate event notification OAMPDUs:   0
                oranization specific OAMPDUs:           0
                link-fault records:                     0
                critical-event records:                 0
                dying-gasp records:                     0
        Rx statistics
                information OAMPDUs:                    682
                loopback control OAMPDUs:               0
                variable request OAMPDUs:               0
                variable response OAMPDUs:              0
                unique event notification OAMPDUs:      0
                duplicate event notification OAMPDUs:   0
                oranization specific OAMPDUs:           0
                unsupported OAMPDUs:                    0
                link-fault records:                     0
                critical-event records:                 0
                dying-gasp records:                     0
                discarded TLVs:                         0
                unrecognized TLVs:                      0
```

Syntax: show link-oam statistics detail ports [all | ethernet <slot/port>]

This field...	Displays...
Ethernet Port	Indicates if the ethernet port that EFM-OAM is enabled on.
Link Status	Indicates if the physical link is operational or any fault is detected on the link.
OAM Status	Indicates the status of OAM on the link between the local and remote DTEs. The status is enabled if OAM client is satisfied with local and remote settings.
Mode	Indicates if the DTE is in active or passive modes. Active DTEs can start the discovery process and passive ones can only respond.
Local Stable	Indicates the reception of the remote DTE state information and is satisfied with the remote OAM settings.
Remote Stable	Indicates the reception of the local DTE state information at the remote DTE and is satisfied with the local OAM settings.

Ping

Ping is a tool that helps you to verify the Internet connectivity at the IP level. The **ping** command sends an Internet Control Message Protocol (ICMP) echo request to the IP address or selected hostname.

Executing ping

The **ping** command, in the (Enable) mode, pings another device from the PowerConnect. The PowerConnect supports IP ping, which you can use to test connectivity to remote hosts. Ping sends an echo request packet to an address and waits for a reply.

The device can execute multiple ping commands at the same time. If you can connect to the device via the console, or through an inbound telnet or SSH session, it should be possible to initiate a ping. This applies to all versions of the ping command described below. The device can also resolve multiple DNS queries simultaneously, which allows multiple ping commands with the **hostname** option to be executed at the same time.

To initiate the PowerConnect to ping to a target device with the IP address of 192.22.2.33, enter a command such as the following.

```
NetIron# ping 192.22.2.33
```

Syntax: **ping** <*ip addr*> | <*hostname*> | **vrf** <*instance-name*> [**source** <*ip addr*>] [**count** <*num*>] [**timeout** <*msec*>] [**ttl** <*num*>] [**size** <*byte*>] [**quiet**] [**numeric**] [**no-fragment**] [**verify**] [**data <1-to-4 byte hex>**] [**brief**]

The required parameter is the IP address or the host name of the device.

The **vrf** <*instance-name*> parameter specifies a VPN routing/forwarding instance as the origin of the ping packets.

The **source** <*ip addr*> parameter specifies an IP address to be used as the origin of the ping packets.

The **count** <*num*> parameter specifies how many ping packets the device sends. You can specify from 1 – 4294967296. The default is 1.

The **timeout** <*msec*> parameter specifies how many milliseconds the device waits for a reply from the pinged device. You can specify a timeout from 1 – 4294967296 milliseconds. The default is 5000 (5 seconds).

The **ttl** <*num*> parameter specifies the maximum number of hops. You can specify a TTL from 1 – 255. The default is 64.

The **size** <*byte*> parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 – 9170. The default is 16.

The **no-fragment** parameter turns on the “do not fragment” bit in the IP header of the ping packet. This option is disabled by default.

The **quiet** parameter hides informational messages such as a summary of the ping parameters sent to the device and instead displays only messages indicating the success or failure of the ping. This option is disabled by default.

The **verify** parameter verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.

The **data** <*1 – 4 byte hex*> parameter lets you specify a specific data pattern for the payload instead of the default data pattern, “abcd”, in the packet’s data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE
For numeric parameter values, the CLI does not check that the value you enter is within the allowed range. If you exceed the range for a numeric value, the software rounds the value to the nearest valid value.

The **brief** keyword causes ping test characters to be displayed. The following ping test characters are supported:

! Indicates that a reply was received.

. Indicates that the network server timed out while waiting for a reply.

U Indicates that a destination unreachable error PDU was received.

I Indicates that the user interrupted ping.

Executing ping VRF

NOTE
The Ping utilities have been enhanced by adding the **ping vrf** command in release 02.1.00 to help with management of Layer-3 VPNs.

The **ping vrf** command lets you test a specific VPN connection. To use this option, enter the following command.

Syntax: ping vrf <*vrf-name*> <*ip-address*>

The **vrf-name** parameter is the name of the VRF that you want to conduct a ping to.

The **ip-address** parameter is the IP address containing the VRF that you want to conduct a ping to.

Executing ping IPv6

The **ping ipv6** command allows you to verify the connectivity from a device to an IPv6 device by performing an ICMP for IPv6 echo test. As with IPv4, multiple IPv6 ping commands can be executed simultaneously by the device.

For example, to ping a device with the IPv6 address of 2001:3424:847f:a385:34dd::45 from the device, enter the following command.

```
NetIron# ping ipv6 2001:3424:847f:a385:34dd::45
```

Syntax: ping ipv6 <*ipv6-address*> | <*host-name*> | **vrf** <*instance-name*>**[outgoing-interface [eth** <*slot/port*> | **pos** <*slot/port*> | **ve** <*number*>**]] [source** <*ipv6-address*>**] [count** <*number*>**] [timeout** <*milliseconds*>**] [ttl** <*number*>**] [size** <*bytes*>**] [quiet] [numeric] [no-fragment] [verify] [data <1-to-4 byte hex>] [brief]**

The required parameter is the IPv6 address or the host name of the device. The <*ipv6-address*> parameter specifies the address of the target device. You must specify this address in hexadecimal using 16-bit values between colons, or specify a host name using an ASCII string.

The **vrf** <*instance-name*> parameter specifies a VPN routing/forwarding instance as the origin of the ping packets.

The **outgoing-interface** keyword specifies a physical interface over which you can verify connectivity. If you specify a physical interface, such as an Ethernet interface, you must also specify the port number of the interface. If you specify a virtual interface, such as a VE, you must specify the number associated with the VE.

Specify either **eth** <*slot/port*> or **pos** <*slot/port*>.

The **source** <*ipv6-address*> parameter specifies an IPv6 address to be used as the origin of the ping packets.

The **count** <*number*> parameter specifies how many ping packets the sends. You can specify from 1 - 4294967296. The default is 1.

The **timeout** <*milliseconds*> parameter specifies how many milliseconds the waits for a reply from the pinged device. You can specify a timeout from 1 - 4294967296 milliseconds. The default is 5000 (5 seconds).

The **ttl** <*number*> parameter specifies the maximum number of hops. You can specify a TTL from 1 - 255. The default is 64.

The **size** <*bytes*> parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 - 9150. The default is 16.

The **no-fragment** keyword turns on the "don't fragment" bit in the IPv6 header of the ping packet. This option is disabled by default.

The **quiet** keyword hides informational messages such as a summary of the ping parameters sent to the device and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.

The **verify** keyword verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.

The **data** <*1 - 4 byte hex*> parameter lets you specify a specific data pattern for the payload instead of the default data pattern, "abcd", in the packet's data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE
For parameters that require a numeric value, the CLI does not check that the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

The **brief** keyword causes ping test characters to be displayed. The following ping test characters are supported:

! Indicates that a reply was received.

. Indicates that the network server timed out while waiting for a reply.

U Indicates that a destination unreachable error PDU was received.

I Indicates that the user interrupted ping.

Trace route

The trace route tool works by sending ICMP echo packets with varying IP Time-to-Live (TTL) values to the destination.

You can use IP traceroute to identify the path that packets take through the network on a hop-by-hop basis. The command output displays all network layer devices, such as routers, through which the traffic passes on its way to the destination.

The device can execute simultaneous **traceroute** commands from multiple inbound telnet or SSH sessions. Multiple simultaneous traceroutes from Web and SNMP, however are not allowed. The device can also resolve multiple DNS queries simultaneously, which allows multiple **traceroute** commands with the **hostname** option to be executed at the same time.

NOTE
Traceroute commands in outbound telnet sessions run on the remote telnet server and not on the local device.

Executing traceroute

The **traceroute** command, in the (Enable) mode, displays the routing path from the routing switch to the destination IP address as soon as the information is received. Traceroute requests display all responses to a given TTL. In addition, if there are multiple equal-cost routes to the destination, the device displays up to three responses by default.

Example

```
NetIron> traceroute 192.33.4.7 minttl 5 maxttl 5 timeout 5
```

Example

```
NetIron1# traceroute vrf blue 10.10.10.10
```

Syntax: **traceroute** <host-ip-addr> [maxttl <value>] [minttl <value>] [numeric] [timeout <value>] [source-ip <ip addr>]

The **maxttl** parameter is the maximum TTL (hops) value: Possible value is 1 – 255. The default is 30 seconds.

The **minttl** parameter is the minimum TTL (hops) value: Possible value is 1 – 255. The default is 1 second.

The **numeric** parameter lets you change the display to list devices by IP address instead of by name.

The **timeout** parameter specifies the possible values. Possible value range is 1 – 120. Default value is 2 seconds.

The **source-ip** <*ip addr*> parameter specifies an IP address to be used as the origin for the traceroute.

The **vrf** <*vrf-name*> parameter is the name of the VRF whose route you want to trace.

The **vrf** <*ip-address*> parameter is the IP address containing the VRFwhose route you want to trace.

Executing traceroute VRF

In the (Enable) mode, the **traceroute vrf** command functions like the standard **traceroute** command but requires you to specify a VRF table name. The **traceroute vrf** command must be used when the route to the destination is associated with a VRF table.

Example

```
NetIron# traceroute vrf blue 10.10.10.10
```

Syntax: traceroute vrf <*vrf-name*> <*ip-address*>

The **vrf** <*vrf-name*> parameter is the name of the VRF for you want are running the traceroute.

The **vrf** <*ip-address*> parameter is the IP address containing the VRF that you want to conduct a traceroute to.

Executing traceroute IPv6

The **traceroute ipv6** command traces a path from a device that supports IPv6 to an IPv6 host.

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a minimum TTL of 1 second and a maximum TTL of 30 seconds. In addition, if there are multiple equal-cost routes to the destination, the device displays up to three responses.

Example

To trace the path from the device to a host with an IPv6 address of 3301:23dd:349e:a384::34, enter the following command.

```
NetIron> traceroute ipv6 3301:23dd:349e:a384::34
```

Syntax: traceroute ipv6 <*ipv6-address*>

The <*ipv6-address*> parameter specifies the address of an IPv6 host. You must specify this address in hexadecimal using 16-bit values between colons.

Trace-l2 protocol

Trace-l2 traces introduces a new proprietary protocol that traces the traffic path to a specified device in a VLAN. Also, it can be used to probe all reachable paths to all devices in a VLAN. It does the following:

- Traces a particular IP, MAC or hostname in a VLAN.
- Probes the entire Layer 2 topology.
- Displays the input or output ports of each hop in the path.
- Displays the round trip travel time of each hop.
- Displays hops in a VLAN that form a loop.
- Displays each hop's Layer 2 protocol such as STP, RSTP, 802.1w, SSTP, metro ring, or route-only.

The resulting trace displays a report that provides information about a packet's path to a device, such as hop and port information and travel time. It also can locate any Layer 2 loop in a VLAN. The probed Layer 2 information is discarded after 10 minutes or when a new **trace-l2** command is issued again.

For each hop in the path, trace-l2 displays its input/output port, L2 protocols of the input port, and the microsecond travel time between hop and hop. It also prints out the hops which form a loop, if any. Displaying L2 topology lets a user easily obtain information of all hops.

Configuration considerations

The configuration considerations are as follows:

- Trace-l2 is enabled on the PowerConnect devices. It can be used to trace traffic only to devices.
- The devices that will participate in the trace-l2 protocol must be assigned to a VLAN and all devices on that VLAN must be Dell devices that support the trace-l2 protocol.
- Dell devices, as well as other vendor devices, that do not support the trace-l2 protocol, simply forward trace-l2 packets without a reply. Hence, these devices are transparent to the trace-l2 protocol.
- The destination for the packet with the trace-l2 protocol must be a device that supports the trace-l2 protocol and the destination cannot be a client, such as a personal computer, or devices from other vendors.

Tracing a traffic path

The trace-l2 protocol is enabled on a VLAN. You can trace the traffic path of a packet by entering a command such as the following.

```
NetIron(config)#trace-l2 vlan 10 2.2.2.2
```

Syntax: [no] trace-l2 vlan <*vlan-id*> <*destination-address*>

The <*destination address*> can be a MAC address, an IP address, or a host. You can enter the destination-address in one of the following formats:

- HHHH.HHHH.HHHH – Destination MAC address
- A.B.C.D – Destination IP address
- ASCII string – destination host name

If a destination address is not specified or the destination does not exists, trace-l2 collects L2 topology information which can be displayed by issuing a **trace-l2 show** command. The command displays the following information.

```
trace-l2 reply vlan 2 from e26, 1.1.1.2, total round trip = 814 microsec
hop input  output   IP and/or MAC address       microsec   comment
  1   e28    e25    1.1.1.4 00e0.803f.c400         316     e28: ring 11
  2   e15    e13    1.1.1.1 0004.8057.0d00         235     e15: ring 11
  3   e27           1.1.1.2 00e0.8057.2500         263     e27: ring 11
```

In the output above, the last hop is the destination. Because 1.1.1.2 and 2.2.2.2 are addresses of the same device, the device can use 1.1.1.2 in the reply.

In general, **trace-l2** first tries to use the IP address of the virtual routing interface that is associated with a VLAN. If the virtual routing interface is not available, it then uses the loopback address. If both addresses are not available, it displays MAC address only.

The **input** and **output** ports show the path of the hops. Hop 3 has no output port because it is the destination.

The **microsec** column is the round trip time (sum of the time) to and from the previous hop. For example, 316 microsec for hop 1 is the time from the source to hop 1 and from hop 1 to the source. One way time is not available because the tracel2 protocol does not synchronize the clocks between hops.

The **comment** column shows the Layer 2 protocol used on the input port. It could be the following:

- STP – spanning tree protocol
- RSTP – Rapid STP, 802.1w draft 3
- 802.1w – Rapid STP

- ring – Metro ring ID of input port.
- Single STP – Includes Single STP, Single RSTP and Single 802.1w
- STP port disabled – The **spanning-tree ethernet disabled** command is configured.
- route-only – This device has route-only configuration
- port route-only – The input port has route-only configuration

Displaying Layer 2 topology information

To display information about the Layer 2 topology, first issue a **trace-l2 vlan** command, then enter the **trace-l2 show** command as in the following example.

```
NetIron(config)#trace-l2 vlan 10
Vlan 10 L2 topology probed, use "trace-l2 show" to display
FES Switch(config)#trace-l2 show
Vlan 10 L2 topology was probed 6 sec ago, # of paths: 2
path 1 from e27, 1 hops:
hop input  output IP and/or MAC address       microsec comment
1   e13           1.1.1.1 0004.8057.0d00           383 802-1w
path 2 from e25, 2 hops:
hop input  output IP and/or MAC address       microsec comment
1   e27    e26    1.1.1.3 00e0.8052.ea00           657 802-1w
2   e28           1.1.1.4 00e0.803f.c400           296 route-only
```

The **trace-l2 show** command does not display a path if the path is a subset of another path; therefore, the number of paths displayed could be fewer than the number of devices.

If the topology contains Layer 2 loops, a message such as the following is displayed.

```
*** Warning! The following 3 hops form a loop in vlan 2
  hop input  output IP and/or MAC address       microsec comment
  1   e25           1.1.2.2 00e0.8057.2500
  2   e28           4.4.100.1 00e0.803f.c400
  3   e29           1.1.1.1 0004.8057.0d00
```

Syntax: trace-l2 show

LSP ping and traceroute

Overview

The LSP Ping and Traceroute feature provides Operation, Administration, and Maintenance (OAM) functionality for MPLS networks based up RFC 4379 (Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures).

The LSP ping and traceroute functions provide a mechanism to detect MPLS data plane failure. LSP ping is used to detect data plane failure and to check the consistency between the data plane and the control plane. LSP traceroute is used to isolate the data plane failure to a particular router and to provide LSP path tracing. They are implemented using MPLS echo request and reply messages which are sent as UDP packets to a well-known UDP port 3503. This section provides the details of LSP Ping and Traceroute operation

LSP ping operation

An MPLS echo request (described in "MPLS echo request" on page 2230) is sent from the ingress to the egress LSR. At the transit LSRs, the ping packet is label switched (the same as a regular MPLS data packet) without any control plane intervention. Upon arriving at the egress LSR, the echo request is sent to the control plane for processing based on the IP Router Alert option and the well-known destination UDP port 3503. An echo reply (described in "MPLS echo reply" on page 2231) is sent back as a UDP packet with an appropriate return code that depends on the result of the FEC stack validation.

LSP traceroute operation

An MPLS echo request (described in "MPLS echo request" on page 2230) is sent from the ingress LSR with the TTL of the outermost label set to an incremental value that starts with a TTL value of 1. This request causes the MPLS echo request to be forwarded to the control plane for processing at each transit LSR, based on the MPLS TTL expiration value. An echo reply (described in "MPLS echo reply" on page 2231) is sent back with a return code indicating that it is the transit LSR for the FEC specified in the echo request. This process repeats until the echo request arrives at the egress LSP. The echo request is then forwarded to the control plane for processing, based on the IP Router Alert option. An echo reply is sent back as a UDP packet with an appropriate return code that depends on the result of the FEC stack validation.

MPLS echo request

The MPLS echo request is sent from the ingress LSR as a labeled UDP packet (except for single-hop LSP). The echo request has the following characteristics.

IP/UDP header information:

- Source address = user-input or LSR ID.
- Destination address = user-input or 127.0.0.1.
- UDP source port = 3503.
- UDP destination port = 3503.
- IP TTL = 1
- Router Alert option is set.

By default, the reply mode is set to 2 (reply by way of an IPv4 UDP packet), and you can set it to 1 (no reply) or 3 (reply by way of an IPv4 UDP packet with Router Alert option).

The sender handle is set to an internally-generated, 32-bit number that is assigned to each ping or traceroute session when the ping or traceroute operation begins. This sender handle is sent back in the echo reply, which is used to locate the appropriate ping or traceroute session.

The sequence number is a running number associated with each ping or traceroute session. It is starts with a value of 1.

The TTL for the outermost label is set to 255 for a ping. For traceroute, it is 1, 2, 3, and so on.

You can configure a timeout when starting the ping or traceroute command. The default value is 5 seconds.

MPLS echo reply

The MPLS echo reply is sent by the transit (for traceroute) or egress (for ping and traceroute) LSR as a regular IPv4 UDP packet or an IPv4 UDP packet with Router Alert option depending on the reply-mode field of the echo request. If reply with Router Alert option is chosen, the user should make sure that all intermediate routers are capable of handling MPLS echo reply. If a reply is sent with Router Alert option and the reply is sent over a tunnel interface, the MPLS Router Alert label (label value 1) will be the topmost label for the packet. A reply with a Router Alert option should be used if and only if the normal IP return path is deemed unreliable.

The echo reply has the following characteristics.

IP/UDP header information:

- Source address = LSR ID
- Destination address = source IP address from the echo request
- UDP source port = 3503
- UDP destination port = UDP source port from the echo request
- IP TTL = 255
- Router Alert option set if and only if reply-mode field of the echo request set to 3.

The sender handle is copied from echo request message

The sequence number is copied from echo request message

LSP ping TLVs

Table 429 lists the TLVs defined in RFC 3479 that are included in an echo request and reply.

TABLE 429 Show Cfm output descriptions

TLV type	TLV name	TX in echo request	TX in echo reply
1	Target FEC stack	Yes	No
2	Downstream mapping	Yes if the **dsmap** option is set	Yes for transit LSRs only if downstream mapping TLV is included in the MPLS Echo request.
3	Pad	Depend on the size option	Yes (if value = 2)
7	Interface and Label Stack	N/A	Yes if the I flag in DS mapping is set
9	Errored TLV	N/A	Yes (if error is detected)
10	Reply TOS bytes	Yes if reply-tos option is set	TLV is not sent back. Just copy TOS byte into IP header.

The PowerConnect devices support sending and receiving downstream mapping TLVs without multipath information (where the multipath type is always set to 0). Note that the detailed multipath information can be used by the ingress LSR to ping or traceroute through all ECMP paths at the transit LSR. Currently, the PowerConnect devices do not support LDP LSPs with ECMP. Consequently, the multipath type of non-zero is not relevant in these operations.

LSP FEC types

For LDP LSPs, the LDP IPv4 prefix sub-TLV (sub-type = 1) is encoded in the target FEC stack of the echo request. For RSVP LSPs, the RSVP IPv4 LSP sub-TLV (sub-type = 3) is encoded in the target FEC stack.

NOTE
Static RSVP LSPs are no longer supported, so a ping or traceroute for a static LSP is not supported.

Redundant RSVP LSPs

For RSVP LSPs with redundant paths, ping or traceroute on a LSP is performed on the currently active path. For example, if the secondary path is the active path for an LSP, the MPLS echo request packets are sent out on the secondary path's interface.

If the active path changes while a ping or traceroute is in progress, the echo request continues to be sent out on the old active path. This implies that the echo request that was sent after path switchover times out. The user subsequently needs to restart the ping or traceroute.

One-to-one Fast ReRoute (FRR) LSPs

Similar to the redundant LSPs, a ping or traceroute on a one-to-one FRR LSP is performed on the active path. If a path switchover occurs while a ping or traceroute is in-progress, the echo request continues to be sent out on the old active path. This implies that the echo request sent after path switchover will time out.

A user can ping or trace the route of the ingress-originated detour of a one-to-one FRR LSP by specifying the detour parameter. The operation is started only if the detour is operationally up.

FRR bypass LSPs

The LSP ping and traceroute facilities support FRR bypass LSPs. You can ping or trace the protected LSP and bypass tunnel separately.

You can ping or trace the ingress-originated or transit-originated bypass tunnel by specifying either the name of bypass LSP (as you would any regular LSP name) or the entire RSVP session ID (including the tunnel endpoint, the tunnel ID, and the extended tunnel ID).

NOTE
In the current facility backup implementation, the bypass LSP name must be unique in the system (for example, the name cannot be the same as the regular LSP name).

The traceroute output of a backup tunnel depends on the setting of the **propagate-ttl** and **label-propagate-ttl** options. If both **propagate-ttl** and **label-propagate-ttl** options are turned on, the traceroute output shows the detail of the bypass path. If both options are turned off, the bypass path is shown as a single hop. The options should be either both ON or both OFF.

To trace the route of a backup path, the TTL of the bypass and protected labels (if they are not implicit NULL labels) are set as in the following example:

- Both **propagate-ttl** and **label-propagate-ttl** are ON: TTL = 1, 2, 3, and so on, are set for both labels.
- Otherwise: bypass label TTL is set to 255. Protected label TTL is set to 1, 2, 3, and so on.

IP TTL is set to topmost label TTL. Otherwise, it is set to 255.

Transit-originated detour

The user can initiate a ping or traceroute operation on a transit-originated, detour LSP. Because the session name does not uniquely identify a session on a transit LSR, the user needs to specify the entire session ID (including the tunnel endpoint, tunnel ID, and extended tunnel ID) for the detour LSP to which the LSP ping or traceroute command is applied.

LSP reoptimization

If LSP reoptimization happens while the ping or traceroute is operating, the echo request is still sent out on the current LSP instance until the new instance is created. This avoids displaying partial information from the old and new paths if they are different; particularly for a traceroute. Similarly, if the ping or traceroute operation is started while LSP reoptimization is occurring, the LSP label, out interface, and other parameters from the currently up instance will be used.

PHP behavior

Ping is transparent to the penultimate LSR. MPLS and IP TTL operations performed on a ping packet are the same as tor a regular data packet. In the default case where the MPLS TTL is copied into the IP TTL, the echo request packet can arrive at the egress LSR with an IP TTL value greater than 1. Consequently, in this situation, the IP Router Alert option is used to direct the echo request packet to the control plane for ping processing.

For a traceroute operation; if the echo request is received with a downstream mapping TLV, the Implicit Null label is encoded in the Downstream label in the echo reply just like any other label.

Since a PowerConnect router advertises an implicit Null label to its upstream LSR for both LDP and RSVP LSPs, packets that arrive at the egress LSR do not have the tunnel label. For a single-hop LSP, the echo request is sent out from ingress LSR as an unlabeled UDP packet.

Using the LSP ping and Traceroute commands

The following sections described operation of the LSP Ping and Traceroute command:

- “Executing LDP LSP ping”
- “Executing RSVP LSP ping”
- “Executing LDP LSP traceroute”
- “Executing RSVP LSP traceroute”

Executing LDP LSP ping

The LDP LSP ping command, sends an MPLS echo request from the ingress to the egress LSR.

To perform the LDP LSP ping operation, use the following command.

```
NetIron)# ping mpls ldp 22.22.22.22
Send 5 80-byte MPLS Echo Requests for LDP FEC 22.22.22.22/32, timeout 5000 msec
Type Control-c to abort
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max=0/1/1 ms.
NetIron)#
```

Syntax: **ping mpls ldp** <*ip-address*> | <*ip-address/mask-length*> [**count** <*num*>] [**destination** <*ip-address*>] [**detail**] [**reply-mode no-reply | reply-mode router-alert**] [**reply-tos** <*num*>] [**size <bytes**] [**source** <*ip-address*>] [**timeout** <*msec*>]

The **ldp** <*ip-address*> and <*ip-address/mask-length*> variables specify the LDP IPv4 destination prefix and mask length. If the **mask-length** is not specified, the default value is 32.

The **count** option with the <*num*> variable specifies the number of echo requests to send. The default value is 5.

The **destination** option specifies an IP address within the 127/8 subnet. The default address is 127.0.0.1

The **detail** option displays the details of the echo request and reply messages. By default, the display is in the brief mode.

The **reply-mode** option species the reply mode field in the echo request if and only if the user does not want the reply to be sent as an IPv4 UDP packet.

The **no-reply** option can be used to test one-way connectivity.

The **router-alert** option is used when the normal IP return path is unreliable. This option indicates that the reply should be sent as an IPv4 UDP packet with the Router Alert option. This option requires extra overhead processing at each LSR along the return path.

The **reply-tos** option specifies a TOS value between 0 and 254 to be included in the Reply-TOS-byte TLV. This value will be copied to the IP header TOS byte of the echo reply. By default, the reply-tos TLV is not included in the echo request.

NOTE
The last bit of the TOS byte is always 0.

The **size** option specifies that the size of the echo request including the label stack to be sent will be the value of the variable <*bytes*>. The pad TLV is used to fill the echo request message to the specified size. The minimum size is 80 byte for an LDP echo request. The maximum size is the size of the LSP MTU.

The **source** option specifies the IP address of any interface. This address is used as the destination address for the echo reply address. The default address is the LSR ID.

The **timeout** option specifies an interval in milliseconds for the echo request message. The default timeout is 5 seconds. The maximum timeout value is 5 minutes.

Executing RSVP LSP ping

The RSVP ping command in the (enable) mode, sends an MPLS echo request from the ingress to the egress LSR.

To perform the RSVP LSP ping operation, use the following command.

```
NetIron# ping mpls rsvp lsp tomlxe2frr-18
Send 5 92-byte MPLS Echo Requests over RSVP LSP tomlxe2frr-18, timeout 5000 msec
Type Control-c to abort
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max=0/1/5 ms.
NetIron)#
```

Syntax: **ping mpls rsvp lsp** <*lsp-name*> **| session** <*tunnel-source-address*> <*tunnel-destination-address*> <*tunnel-id*> **[count** <*num*>**] [destination** <*ip-address*>**] [detail] [detour] [reply-mode no-reply | reply-mode router-alert] [reply-tos** <*num*>**] [size <bytes] [source** <*ip-address*>**] [standby] [timeout** <*msec*>**]**

The **rsvp lsp** option specifies the name of the RSVP IPv4 LSP in the <*lsp-name*> variable.

The **rsvp session** option specifies the session ID. The <*tunnel-source-address*>, <*tunnel-destination-address*> and <*tunnel-id*> variables must all be specified to form a valid session ID.

The **count** option with the <*num*> variable specifies the number of echo requests to send. The default value is 5.

The **destination** option specifies an IP address within the 127/8 subnet. The default address is 127.0.0.1

The **detail** option displays the details of the echo request and reply messages. By default, the display is in the brief mode.

The **detour** option specifies a ping detour path. For a detour originated on the ingress LSR, you can ping the detour path using either the LSP name or session ID with the **detour** option specified.

The **reply-mode** option species the reply mode field in the echo request if and only if the user does not want the reply to be sent as an IPv4 UDP packet.

The **no-reply** option can be used to test one-way connectivity.

The **router-alert** option is used when the normal IP return path is unreliable. This option indicates that the reply should be sent as an IPv4 UDP packet with the Router Alert option. This option requires extra overhead processing at each LSR along the return path.

The **reply-tos** option specifies a TOS value between 0 and 254 to be included in the Reply-TOS-byte TLV. This value will be copied to the IP header TOS byte of the echo reply. By default, the reply-tos TLV is not included in the echo request.

NOTE
The last bit of the TOS byte is always 0.

The **size** option specifies that the size of the echo request including the label stack to be sent will be the value of the variable <*bytes*>. The pad TLV is used to fill the echo request message to the specified size. The minimum size is 92 bytes for an MPLS Echo request. The maximum size is the size of the LSP MTU.

The **source** option specifies the IP address of any interface. This address is used as the destination address for the echo reply address. The default address is the LSR ID.

The **standby** option directs the ping operation to the secondary path of a redundant LSP that is operationally up.

The **timeout** option specifies an interval in milliseconds for the echo request message. The default timeout is 5 seconds. The maximum timeout value is 5 minutes.

Executing LDP LSP traceroute

The LDP LSP traceroute command in the (enable) mode, sends and MPLS echo request from the ingress to the egress LSR.

To perform the LDP LSP traceroute operation, use the following command.

```
NetIron# traceroute mpls ldp 22.22.22.22
Trace LDP LSP to 22.22.22.22/32, timeout 5000 msec, TTL 1 to 30
Type Control-c to abort
 1 10ms 22.22.22.22 return code 3(Egress)
NetIron)#
```

Syntax: **traceroute mpls ldp** <*ip-address/mask-length*> **[destination** <*ip-address*>**] [dsmap] [min-ttl** <*min-num*>**] [max-ttl** <*max-num*>**] [reply-mode router-alert] [reply-tos** <*num*>**] [size <bytes] [source** <*ip-address*>**] [timeout** <*msec*>**]**

The **ldp** <*ip-address/mask-length*> variable specifies the LDP IPv4 destination prefix and mask length. If the **mask-length** is not specified, the default value is 32.

The **destination** option specifies an IP address within the 127/8 subnet. The default address is 127.0.0.1

The **dsmap** option enables the Downstream (DS) mapping TLV in the echo request for traceroute operation. The DS mapping TLV is used to instruct the transit LSR to include the next-hop interface and label information in the echo reply. By default, the DS TLV is not included in the echo request.

The **min-ttl** option specifies a minimum value in the <*min-num*> variable for the outermost label in traceroute operation. The default minimum TTL value is 1. Acceptable values that can be configured are: 1 - 255.

The **max-ttl** option specifies a maximum value in the <*max-num*> variable for the outermost label in traceroute operation. The default maximum TTL value is 30. Acceptable values that can be configured are: 1 - 255.

The **reply-mode router-alert** option is used when the normal IP return path is unreliable. This option indicates that the reply should be sent as an IPv4 UDP packet with the Router Alert option. This option requires extra overhead processing at each LSR along the return path.

The **reply-tos** option specifies a TOS value between 0 and 254 to be included in the Reply-TOS-byte TLV. This value will be copied to the IP header TOS byte of the echo reply. By default, the reply-tos TLV is not included in the echo request.

NOTE
The last bit of the TOS byte is always 0.

The **size** option specifies that the size of the echo request including the label stack to be sent will be the value of the variable <*bytes*>. The pad TLV is used to fill the echo request message to the specified size. The minimum size is 92 bytes for an MPLS Echo request. The maximum size is the size of the LSP MTU.

The **source** option specifies the IP address of any interface. This address is used as the destination address for the echo reply address. The default address is the LSR ID.

The **timeout** option specifies an interval in milliseconds for the echo request message. The default timeout is 5 seconds. The maximum timeout value is 5 minutes.

Executing RSVP LSP traceroute

The RSVP LSP traceroute command in the (enable) mode, sends and MPLS echo request from the ingress to the egress LSR.

To perform the RSVP LSP traceroute operation, use the following command.

```
NetIron # traceroute mpls rsvp lsp tomlxe2frr-18
Trace RSVP LSP tomlxe2frr-18, timeout 5000 msec, TTL 1 to 30
Type Control-c to abort
 1 1ms 22.22.22.22 return code 3(Egress)
NetIron#
```

Syntax: **traceroute mpls rsvp lsp** <*lsp-name*> | **session** <*tunnel-source-address*> <*tunnel-destination-address*> <*tunnel-id*> **[destination** <*ip-address*>**] [dsmap] [detour] [min-ttl** <*min-num*>**] [max-ttl** <*max-num*>**] [reply-mode router-alert] [reply-tos** <*num*>**] [size <bytes] [source** <*ip-address*>**] [standby] [timeout** <*msec*>**]**

The **rsvp lsp** option specifies the name of the RSVP IPv4 LSP in the <*lsp-name*> variable.

The **rsvp session** option specifies the session ID. The <*tunnel-source-address*>, <*tunnel-destination-address*> and <*tunnel-id*> variables must all be specified to form a valid session ID.

The **destination** option specifies an IP address within the 127/8 subnet. The default address is 127.0.0.1

The **dsmap** option enables the Downstream (DS) mapping TLV in the echo request for traceroute operation. The DS mapping TLV is used to instruct the transit LSR to include the next-hop interface and label information in the echo reply. By default, the DS TLV is not included in the echo request.

The **detour** option specifies a traceroute detour path. For a detour originated on the ingress LSR, you can ping the detour path using either the LSP name or session ID with the **detour** option specified.

The **reply-mode router-alert** option is used when the normal IP return path is unreliable. This option indicates that the reply should be sent as an IPv4 UDP packet with the Router Alert option. This option requires extra overhead processing at each LSR along the return path.

The **reply-tos** option specifies a TOS value between 0 and 254 to be included in the Reply-TOS-byte TLV. This value will be copied to the IP header TOS byte of the echo reply. By default, the reply-tos TLV is not included in the echo request.

NOTE
The last bit of the TOS byte is always 0.

The **size** option specifies that the size of the echo request including the label stack to be sent will be the value of the variable <*bytes*>. The pad TLV is used to fill the echo request message to the specified size. The minimum size is 92 bytes for an MPLS Echo request. The maximum size is the size of the LSP MTU.

The **source** option specifies the IP address of any interface. This address is used as the destination address for the echo reply address. The default address is the LSR ID.

The **standby** option directs the traceroute operation to the secondary path of a redundant LSP that is operationally up.

The **timeout** option specifies an interval in milliseconds for the echo request message. The default timeout is 5 seconds. The maximum timeout value is 5 minutes.

Displaying LSP ping and traceroute statistics

You can use the **show mpls statistics oam** command to display the following LSP ping and traceroute counters:

- Ping and traceroute requests that are issued by the user
- Echo requests sent
- Echo requests received
- Echo request time-outs
- Echo replies sent
- Echo replies received
- Echo replies with error return codes

To display the LSP ping and traceroute counters use the **show mpls statistics oam** command, as shown in the following.

```
PowerConnect # show mpls statistics oam
User ping request processed: 8
User traceroute request processed: 3
Echo requests: sent(102658), received(2865), timeout(0)
Echo replies: sent(2865), received(102628)
Echo reply return code distribution:        TX         RX
  Egress(3)                        :        0          102628
  Transit(8)                       :        0          0
  No return code(0)                :        0          0
  Malformed request(1)             :        0          0
  Unsupported TLV(2)               :        2865       0
  No FEC mapping(4)                :        0          0
  DS map mismatch(5)               :        0          0
  Unknown upstream intf(6)         :        0          0
  Reserved return code(7)          :        0          0
  Unlabeled output intf(9)         :        0          0
  FEC mapping mismatch(10)         :        0          0
  No label entry(11)               :        0          0
  Rx intf protocol mismatch(12)    :        0          0
  Premature LSP termination(13)    :
```

Syntax: show mpls statistics oam

When the detail option is specified, the echo reply is shown with a error return code based on the error codes listed in RFC 4379.

Clearing the LSP ping and traceroute counters

You can use the **clear mpls statistics oam** command to clear the LSP ping and traceroute counters as shown in the following.

```
NetIron # clear mpls statistics oam
```

Syntax: clear mpls statistics oam

Chapter

Foundry Direct Routing and CAM Partition Profiles for the PowerConnect B-MLXe 61

PowerConnect B-MLXe supports the following Foundry Direct Routing (FDR) and CAM Partition features:

- Foundry Direct Routing
- CAM Partition Profiles
- Supernet CAM Partition Sharing
- CAM Overflow Logging

Configuring FDR globally

The default CAM modecurrently supported is static CAM mode which is also known as Foundry Direct Routing (FDR). You can set the CAM mode to dynamic IP CAM mode using the following command:

```
NetIron(config)# cam-mode ip dynamic
```

You must reload the device for this command to take effect.

Syntax: [**no**] **cam-mode ip dynamic** [**dynamic** | **static**]

The **dynamic** parameter sets the IP CAM mode to dynamic.

The **static** parameter sets the IP CAM mode to static. This is the default state.

Configuring FDR for IPv6 routes

FDR, also known as IP static CAM mode, can be configured for IPv6 routes. The default for IPv6 routes is static CAM mode. You can set the CAM mode to dynamic using the following command:

```
NetIron(config)# cam-mode ipv6 dynamic
```

You must reload the device for this command to take effect.

Syntax: [**no**] **cam-mode ipv6** [**dynamic** | **static** | **host**]

The **dynamic** parameter sets the IPv6 CAM mode to dynamic, and programs only the IPv6 prefix into the CAM.

The **static** parameter sets the IPv6 CAM mode to static, and programs only the IPv6 prefix into the CAM. This is the default state.

The **host** parameter programs the complete 128 bit IPv6 address into the CAM.

Configuring FDR for IPv4 and IPv6 VPN routes

FDR can be configured for IPv4 or IPv6 VPN routes. The default CAM mode for IPv4 or IPv6 VPN routes is static. You can set the CAM mode to dynamic using the following command:

```
NetIron(config)# cam-mode ipvpn dynamic
```

You must reload the device for this command to take effect.

Syntax: [no] cam-mode ipvpn [dynamic | static]

The dynamic parameter sets the IPv4 or IPv6 VPN CAM mode to dynamic.

The static parameter sets the IPv4 or IPv6 VPN CAM mode to static. This is the default state.

CAM partition profiles

CAM is partitioned on the device by a variety of profiles that you can select depending on your application.

To implement a CAM partition profile, enter the following command.

```
NetIron(config) cam-partition profile ipv4
```

Syntax: cam-partition profile [ipv4 | ipv4-ipv6| ipv4-ipv6-2 | ipv4-vpls | ipv4-vpn | ipv6 | l2-metro | l2-metro-2 | mpls-l3vpn | mpls-l3vpn-2 | mpls-vpls | mpls-vpls-2 | mpls-vpn-vpls | multi-service | multi-service-2 |multi-service-3 | multi-service-4]

The ipv4 parameter adjusts the CAM partitions, optimize the device for IPv4 applications.

The ipv4-ipv6 parameter adjusts the CAM partitions, to optimize the device for IPv4 and IPv6 dual stack applications

The ipv4-ipv6-2 parameter adjusts the CAM partitions, to optimize the device for increased IPv4 routes with room or IPv6.

The ipv4-vpls parameter adjusts the CAM partitions, to optimize the device for IPv4 and MPLS VPLS applications

The ipv4-vpn parameter adjusts the CAM partitions, to optimize the device for IPv4 and MPLS Layer-3 VPN applications

The ipv6 parameter adjusts the CAM partitions, to optimize the device for IPv6 applications.

The l2-metro parameter adjusts the CAM partitions, to optimize the device for Layer 2 Metro applications.

The l2-metro-2 parameter provides another alternative to l2-metro to optimize the device for Layer 2 Metro applications. It adjusts the CAM partitions.

The mpls-l3vpn parameter adjusts the CAM partitions, to optimize the device for Layer 3, BGP or MPLS VPN applications.

The mpls-l3vpn-2 parameter provides another alternative to mpls-l3vpn to optimize the device for Layer 3, BGP or MPLS VPN applications. It adjusts the CAM partitions.

The mpls-vpls parameter adjusts the CAM partitions, to optimize the device for MPLS VPLS applications.

The mpls-vpls-2 parameter provides another alternative to mpls-vpls to optimize the device for MPLS VPLS applications. It adjusts the CAM partitions.

The mpls-vpn-vpls parameter adjusts the CAM partitions, to optimize the device for MPLS Layer-3 and Layer-2 VPN applications.

The **multi-service** parameter adjusts the CAM partitions, to optimize the device for Multi-Service applications.

The **multi-service-2** parameter provides another alternative to **multi-service** to optimize the device for Multi-Service applications. It adjusts the CAM partitions.

NOTE
You must reload your device for this command to take effect.

The **multi-service-3** parameter provides another alternative to multi-service to optimize the device for Multi-Service applications to support IPv6 VRF. It adjusts the CAM partitions.

The **multi-service-4** parameter provides another alternative to multi-service to optimize the device for Multi-Service applications to support IPv6 VRF. It adjusts the CAM partitions.

Supernet CAM partition sharing

TCAM allocation is optimized to allow dynamic allocation of resources to each level. If one level runs out of TCAM resources, it can use resources that have been allocated to another level and remain unused. This feature is applicable to IPv4, IPv6, and L-3 VPN routes.

Displaying CAM partition

The **show cam-partition** command provides information about available CAM in three formats: raw size, user size, and reserved size.

```
NetIron#show cam-partition
CAM partitioning profile: default
Slot 1 XPP20SP 0:
# of CAM device               = 4
Total CAM Size                = 917504 entries (63Mbits)
IP: Raw Size 524288, User Size 524288(0 reserved)
  Subpartition  0: Raw Size 12288, User Size 12288, (0 reserved)
  Subpartition  1: Raw Size 468107, User Size 468107, (0 reserved)
  Subpartition  2: Raw Size 37335, User Size 37335, (0 reserved)
  Subpartition  3: Raw Size 5140, User Size 5140, (0 reserved)
  Subpartition  4: Raw Size 778, User Size 778, (0 reserved)
IPv6: Raw Size 131072, User Size 65536(0 reserved)
  Subpartition  0: Raw Size 12288, User Size 6144, (0 reserved)
  Subpartition  1: Raw Size 107496, User Size 53748, (0 reserved)
  Subpartition  2: Raw Size 9332, User Size 4666, (0 reserved)
  Subpartition  3: Raw Size 1284, User Size 642, (0 reserved)
  Subpartition  4: Raw Size 384, User Size 192, (0 reserved)
IP VPN Raw Size 131072, User Size 131072(0 reserved)
  Subpartition  0: Raw Size 2048, User Size 2048, (0 reserved)
  Subpartition  1: Raw Size 116886, User Size 116886, (0 reserved)
  Subpartition  2: Raw Size 9333, User Size 9333, (0 reserved)
  Subpartition  3: Raw Size 1285, User Size 1285, (0 reserved)
  Subpartition  4: Raw Size 384, User Size 384, (0 reserved)
MAC: Raw Size 131072, User Size 131072(0 reserved)
  Subpartition  0: Raw Size 10, User Size 10, (0 reserved)
  Subpartition  1: Raw Size 32, User Size 32, (0 reserved)
  Subpartition  2: Raw Size 131030, User Size 131030, (0 reserved)
Session: Raw Size 98304, User Size 49152(0 reserved)
  Subpartition  0: Raw Size 79872, User Size 39936, (0 reserved)
  Subpartition  1: Raw Size 2048, User Size 1024, (0 reserved)
  Subpartition  2: Raw Size 16384, User Size 8192, (0 reserved)
```

```
IPv6 Session: Raw Size 32768, User Size 4096(0 reserved)
  Subpartition  0: Raw Size 15872, User Size 1984, (0 reserved)
  Subpartition  1: Raw Size 512, User Size 64, (0 reserved)
  Subpartition  2: Raw Size 16384, User Size 2048, (0 reserved)
Out Session: Raw Size 196608, User Size 98304(49152 reserved)
Out IPv6 Session: Raw Size 65536, User Size 8192(4096 reserved)
Slot 1 XPP20SP 0:
IP Section(Left):      0(000000) - 262143(03ffff)
IP Section(Right:      0 (000000) - 262143 (03ffff)
  IP SNet  0:(Left):      0(000000) -  12287(002fff)
  IP SNet  1:(Left):  12288(003000) - 262143(03ffff)
  IP SNet  1:(Right):      0 (000000) - 218250 (03548a)
  IP SNet  2:(Right): 218251 (03548b) - 255585 (03e661)
  IP SNet  3:(Right): 255586 (03e662) - 260725 (03fa75)
  IP SNet  4:(Right): 260726 (03fa76) - 261503 (03fd7f)
  IP SNet  5:(Right): 261504 (03fd80) - 261631 (03fdff)
  IP SNet  6:(Right): 261632 (03fe00) - 261695 (03fe3f)
  IP SNet  7:(Right): 261696 (03fe40) - 261727 (03fe5f)
  IP SNet  8:(Right): 261728 (03fe60) - 261759 (03fe7f)
  IP SNet  9:(Right): 261760 (03fe80) - 261791 (03fe9f)
  IP SNet 10:(Right): 261792 (03fea0) - 261807 (03feaf)
  IP SNet 11:(Right): 261808 (03feb0) - 261823 (03febf)
  IP SNet 12:(Right): 261824 (03fec0) - 261839 (03fecf)
  IP SNet 13:(Right): 261840 (03fed0) - 261855 (03fedf)
  IP SNet 14:(Right): 261856 (03fee0) - 261871 (03feef)
  IP SNet 15:(Right): 261872 (03fef0) - 261887 (03feff)
  IP SNet 16:(Right): 261888 (03ff00) - 261903 (03ff0f)
  IP SNet 17:(Right): 261904 (03ff10) - 261919 (03ff1f)
  IP SNet 18:(Right): 261920 (03ff20) - 261935 (03ff2f)
  IP SNet 19:(Right): 261936 (03ff30) - 261951 (03ff3f)
  IP SNet 20:(Right): 261952 (03ff40) - 261967 (03ff4f)
  IP SNet 21:(Right): 261968 (03ff50) - 261983 (03ff5f)
  IP SNet 22:(Right): 261984 (03ff60) - 261999 (03ff6f)
  IP SNet 23:(Right): 262000 (03ff70) - 262015 (03ff7f)
  IP SNet 24:(Right): 262016 (03ff80) - 262031 (03ff8f)
  IP SNet 25:(Right): 262032 (03ff90) - 262047 (03ff9f)
  IP SNet 26:(Right): 262048 (03ffa0) - 262063 (03ffaf)
  IP SNet 27:(Right): 262064 (03ffb0) - 262079 (03ffbf)
  IP SNet 28:(Right): 262080 (03ffc0) - 262095 (03ffcf)
  IP SNet 29:(Right): 262096 (03ffd0) - 262111 (03ffdf)
  IP SNet 30:(Right): 262112 (03ffe0) - 262127 (03ffef)
  IP SNet 31:(Right): 262128 (03fff0) - 262143 (03ffff)
IPV6 Section    : 262144 (040000) - 393215 (05ffff)
  IPV6 SNet  0: 262144 (040000) - 274431 (042fff)
  IPV6 SNet  1: 274432 (043000) - 381927 (05d3e7)
  IPV6 SNet  2: 381928 (05d3e8) - 391259 (05f85b)
  IPV6 SNet  3: 391260 (05f85c) - 392543 (05fd5f)
  IPV6 SNet  4: 392544 (05fd60) - 392927 (05fedf)
  IPV6 SNet  5: 392928 (05fee0) - 393055 (05ff5f)
  IPV6 SNet  6: 393056 (05ff60) - 393119 (05ff9f)
  IPV6 SNet  7: 393120 (05ffa0) - 393151 (05ffbf)
  IPV6 SNet  8: 393152 (05ffc0) - 393183 (05ffdf)
  IPV6 SNet  9: 393184 (05ffe0) - 393215 (05ffff)
IP VPN Section: 393216 (060000) - 524287 (07ffff)
  IP VPN SNet  0: 393216 (060000) - 395263 (0607ff)
  IP VPN SNet  1: 395264 (060800) - 512149 (07d095)
  IP VPN SNet  2: 512150 (07d096) - 521482 (07f50a)
  IP VPN SNet  3: 521483 (07f50b) - 522767 (07fa0f)
  IP VPN SNet  4: 522768 (07fa10) - 523151 (07fb8f)
  IP VPN SNet  5: 523152 (07fb90) - 523279 (07fc0f)
```

```
  IP VPN SNet  6: 523280 (07fc10) - 523343 (07fc4f)
  IP VPN SNet  7: 523344 (07fc50) - 523375 (07fc6f)
  IP VPN SNet  8: 523376 (07fc70) - 523407 (07fc8f)
  IP VPN SNet  9: 523408 (07fc90) - 523439 (07fcaf)
  IP VPN SNet 10: 523440 (07fcb0) - 523455 (07fcbf)
  IP VPN SNet 11: 523456 (07fcc0) - 523471 (07fccf)
  IP VPN SNet 12: 523472 (07fcd0) - 523487 (07fcdf)
  IP VPN SNet 13: 523488 (07fce0) - 523503 (07fcef)
  IP VPN SNet 14: 523504 (07fcf0) - 523519 (07fcff)
  IP VPN SNet 15: 523520 (07fd00) - 523535 (07fd0f)
  IP VPN SNet 16: 523536 (07fd10) - 523551 (07fd1f)
  IP VPN SNet 17: 523552 (07fd20) - 523567 (07fd2f)
  IP VPN SNet 18: 523568 (07fd30) - 523583 (07fd3f)
  IP VPN SNet 19: 523584 (07fd40) - 523599 (07fd4f)
  IP VPN SNet 20: 523600 (07fd50) - 523615 (07fd5f)
  IP VPN SNet 21: 523616 (07fd60) - 523631 (07fd6f)
  IP VPN SNet 22: 523632 (07fd70) - 523647 (07fd7f)
  IP VPN SNet 23: 523648 (07fd80) - 523663 (07fd8f)
  IP VPN SNet 24: 523664 (07fd90) - 523679 (07fd9f)
  IP VPN SNet 25: 523680 (07fda0) - 523695 (07fdaf)
  IP VPN SNet 26: 523696 (07fdb0) - 523711 (07fdbf)
  IP VPN SNet 27: 523712 (07fdc0) - 523727 (07fdcf)
  IP VPN SNet 28: 523728 (07fdd0) - 523743 (07fddf)
  IP VPN SNet 29: 523744 (07fde0) - 523759 (07fdef)
  IP VPN SNet 30: 523760 (07fdf0) - 523775 (07fdff)
  IP VPN SNet 31: 523776 (07fe00) - 524287 (07ffff)
MAC Section     : 524288 (080000) - 655359 (09ffff)
  MAC Forwarding: 524288 (080000) - 655317 (09ffd5)
  MAC Flooding  : 655318 (09ffd6) - 655327 (09ffdf)
  Misc Protocol : 655350 (09fff6) - 655381 (0a0015)
Session Section : 655360 (0a0000) - 753663 (0b7fff)
  IP Multicast  : 655360 (0a0000) - 671743 (0a3fff)
  Receive ACL   : 671744 (0a4000) - 673791 (0a47ff)
  Rule-based ACL: 673792 (0a4800) - 753663 (0b7fff)
IPv6 Session Sec: 753664 (0b8000) - 786431 (0bffff)
  IP Multicast  : 753664 (0b8000) - 770047 (0bbfff)
  Receive ACL   : 770048 (0bc000) - 770559 (0bc1ff)
  Rule-based ACL: 770560 (0bc200) - 786431 (0bffff)
Out Session     : 786432 (0c0000) - 983039 (0effff)
Out IPv6 Session: 983040 (0f0000) - 104857 (0fffff)
...
```

The formats are:

- **Raw Size:** This is double the value of the CAM partition standard entry count. A standard entry contains 64 bits of data and 64 bits for a mask. The raw size may cover invalid entries.
- **User Size:** The actual number of entries that the application can use. For a 128bit application, such as L4 ACL and IPV6, two standard entries equal one user entry. This format also covers invalid entries.
- **Reserved:** The number of entries not usable in this partition. You can subtract this value from user size to obtain the actual available CAM size.

Syntax: show cam-partition

Displaying CAM Partition for IPv6 VPN

The IPv6 VPN CAM partition is created when multi-service-3 or multi-service-4 CAM profile is configured. The IPv6 VPN CAM partition contains 10 sub partitions.The sub-partition is allocated with a fixed size, but can be dynamically changed. If the size of sub-partition is dynamically changed, the output from the **show cam-partition** command is affected. The following example displays information about IPv6 VPN CAM partition when the current CAM profile is multi-service-3:

```
NetIron# show cam-partition
CAM partitioning profile: multi-service-3
Slot 1 XPP20SP 0:
# of CAM device                = 4
Total CAM Size                 = 917504 entries (63Mbits)
……
IPv6 VPN: Raw Size 131072, User Size 65536(0 reserved)
  Subpartition  0: Raw Size 2048, User Size 1024, (0 reserved)
  Subpartition  1: Raw Size 117734, User Size 58867, (0 reserved)
  Subpartition  2: Raw Size 9333, User Size 4666, (0 reserved)
  Subpartition  3: Raw Size 1285, User Size 642, (0 reserved)
  Subpartition  4: Raw Size 384, User Size 192, (0 reserved)
……
Slot 1 XPP20SP 0:
……
IPv6 VPN Section: 524288 (080000) - 655359 (09ffff)
  IPV6 VPN SNet  0: 524288 (080000) - 526335 (0807ff)
  IPV6 VPN SNet  1: 526336 (080800) - 644069 (09d3e5)
  IPV6 VPN SNet  2: 644070 (09d3e6) - 653402 (09f85a)
  IPV6 VPN SNet  3: 653403 (09f85b) - 654687 (09fd5f)
  IPV6 VPN SNet  4: 654688 (09fd60) - 655071 (09fedf)
  IPV6 VPN SNet  5: 655072 (09fee0) - 655199 (09ff5f)
  IPV6 VPN SNet  6: 655200 (09ff60) - 655263 (09ff9f)
  IPV6 VPN SNet  7: 655264 (09ffa0) - 655295 (09ffbf)
  IPV6 VPN SNet  8: 655296 (09ffc0) - 655327 (09ffdf)
  IPV6 VPN SNet  9: 655328 (09ffe0) - 655359 (09ffff)
```

Syntax: show cam-partition

Output from show CAM partition usage command

The **show cam-partition usage** command shows the CAM size available per partition, the amount free, and the percent used. This informarion is shown here for slot 1.

```
NetIron# show cam-partition usage
CAM partitioning profile: multi-service-3

Slot 1 XPP20SP 0:
Slot 1 XPP20SP 0:
           [IP]262144(size), 262129(free), 00.00%(used)
       :SNet 0:  2048(size),   2036(free), 00.58%(used)
       :SNet 1:237830(size), 237828(free), 00.00%(used)
       :SNet 2: 18667(size),  18667(free), 00.00%(used)
       :SNet 3:  2570(size),   2570(free), 00.00%(used)
       :SNet 4:   389(size),    389(free), 00.00%(used)
       :SNet 5:   128(size),    128(free), 00.00%(used)
       :SNet 6:    64(size),     64(free), 00.00%(used)
       :SNet 7:    32(size),     32(free), 00.00%(used)
```

```
 :SNet 8:    32(size),     32(free), 00.00%(used)
 :SNet 9:    32(size),     32(free), 00.00%(used)
:SNet 10:    16(size),     16(free), 00.00%(used)
:SNet 11:    16(size),     16(free), 00.00%(used)
:SNet 12:    16(size),     16(free), 00.00%(used)
:SNet 13:    16(size),     16(free), 00.00%(used)
:SNet 14:    16(size),     16(free), 00.00%(used)
:SNet 15:    16(size),     16(free), 00.00%(used)
:SNet 16:    16(size),     16(free), 00.00%(used)
:SNet 17:    16(size),     16(free), 00.00%(used)
:SNet 18:    16(size),     16(free), 00.00%(used)
:SNet 19:    16(size),     16(free), 00.00%(used)
:SNet 20:    16(size),     16(free), 00.00%(used)
:SNet 21:    16(size),     16(free), 00.00%(used)
:SNet 22:    16(size),     16(free), 00.00%(used)
:SNet 23:    16(size),     16(free), 00.00%(used)
:SNet 24:    16(size),     16(free), 00.00%(used)
:SNet 25:    16(size),     16(free), 00.00%(used)
:SNet 26:    16(size),     16(free), 00.00%(used)
:SNet 27:    16(size),     16(free), 00.00%(used)
:SNet 28:    16(size),     16(free), 00.00%(used)
:SNet 29:    16(size),     16(free), 00.00%(used)
:SNet 30:    16(size),     16(free), 00.00%(used)
:SNet 31:    16(size),     15(free), 06.25%(used)

   [IPV6] 32768(size),  32762(free), 00.01%(used)
 :SNet 0:  1024(size),   1022(free), 00.19%(used)
 :SNet 1: 28756(size),  28754(free), 00.00%(used)
 :SNet 2:  2332(size),   2332(free), 00.00%(used)
 :SNet 3:   320(size),    320(free), 00.00%(used)
 :SNet 4:   192(size),    192(free), 00.00%(used)
 :SNet 5:    64(size),     64(free), 00.00%(used)
 :SNet 6:    32(size),     32(free), 00.00%(used)
 :SNet 7:    16(size),     16(free), 00.00%(used)
 :SNet 8:    16(size),     15(free), 06.25%(used)
 :SNet 9:    16(size),     15(free), 06.25%(used)
[IP VPN]196608(size), 196532(free), 00.03%(used)
 :SNet 0:  2048(size),   1999(free), 02.39%(used)
 :SNet 1:177113(size), 177086(free), 00.01%(used)
 :SNet 2: 14000(size),  14000(free), 00.00%(used)
 :SNet 3:  1927(size),   1927(free), 00.00%(used)
 :SNet 4:   384(size),    384(free), 00.00%(used)
 :SNet 5:   128(size),    128(free), 00.00%(used)
 :SNet 6:    64(size),     64(free), 00.00%(used)
 :SNet 7:    32(size),     32(free), 00.00%(used)
 :SNet 8:    32(size),     32(free), 00.00%(used)
 :SNet 9:    32(size),     32(free), 00.00%(used)
:SNet 10:    16(size),     16(free), 00.00%(used)
:SNet 11:    16(size),     16(free), 00.00%(used)
:SNet 12:    16(size),     16(free), 00.00%(used)
:SNet 13:    16(size),     16(free), 00.00%(used)
:SNet 14:    16(size),     16(free), 00.00%(used)
:SNet 15:    16(size),     16(free), 00.00%(used)
:SNet 16:    16(size),     16(free), 00.00%(used)
:SNet 17:    16(size),     16(free), 00.00%(used)
:SNet 18:    16(size),     16(free), 00.00%(used)
:SNet 19:    16(size),     16(free), 00.00%(used)
:SNet 20:    16(size),     16(free), 00.00%(used)
:SNet 21:    16(size),     16(free), 00.00%(used)
:SNet 22:    16(size),     16(free), 00.00%(used)
```

```
          :SNet 23:    16(size),     16(free), 00.00%(used)
          :SNet 24:    16(size),     16(free), 00.00%(used)
          :SNet 25:    16(size),     16(free), 00.00%(used)
          :SNet 26:    16(size),     16(free), 00.00%(used)
          :SNet 27:    16(size),     16(free), 00.00%(used)
          :SNet 28:    16(size),     16(free), 00.00%(used)
          :SNet 29:    16(size),     16(free), 00.00%(used)
          :SNet 30:    16(size),     16(free), 00.00%(used)
          :SNet 31:   512(size),    512(free), 00.00%(used)

              [MAC]131072(size), 131061(free), 00.00%(used)
        :Protocol:    10(size),      6(free), 40.00%(used)
      :Forwarding:131054(size), 131047(free), 00.00%(used)
        :Flooding:     8(size),      8(free), 00.00%(used)

        [IPV6 VPN] 65536(size),     15(free), 99.97%(used)
          :SNet 0:    20(size),      0(free), 100.00%(used)
          :SNet 1: 65500(size),      0(free), 100.00%(used)
          :SNet 2:     2(size),      2(free), 00.00%(used)
          :SNet 3:     2(size),      2(free), 00.00%(used)
          :SNet 4:     2(size),      2(free), 00.00%(used)
          :SNet 5:     2(size),      2(free), 00.00%(used)
          :SNet 6:     2(size),      2(free), 00.00%(used)
          :SNet 7:     2(size),      2(free), 00.00%(used)
          :SNet 8:     2(size),      1(free), 50.00%(used)
          :SNet 9:     2(size),      2(free), 00.00%(used)

         [Session] 32768(size),  32767(free), 00.00%(used)
    :IP Multicast:  8192(size),   8192(free), 00.00%(used)
     :Receive ACL:  1024(size),   1023(free), 00.09%(used)
        :Rule ACL: 23552(size),  23552(free), 00.00%(used)
    :IP Source Guard Permit:     0(size),      0(free), 00.00%(used)
    :IP Source Guard Denial:     0(size),      0(free), 00.00%(used)

    [IPV6 Session]  8192(size),   8192(free), 00.00%(used)
    :IP Multicast:  2048(size),   2048(free), 00.00%(used)
     :Receive ACL:     0(size),      0(free), 00.00%(used)
        :Rule ACL:  6144(size),   6144(free), 00.00%(used)

      [Internal Forwarding Lookup]  8192(size),   8185(free), 00.08%(used)

     [Out Session] 28672(size),  28672(free), 00.00%(used)

  [Out V6 Session]  8192(size),   8192(free), 00.00%(used)
```

The type of CAM partitioning profile configured is displayed in the “CAM partitioning profile line. The “multi-service-3” or “multi-service-4” provife indicates that the system will allocate a partition for IPV6 VPN.

The output displays the size of the available CAM, amount of CAM currently free, and what percentage of the available CAM is used currently.

(**size**): The effective user size obtained by subtracting the reserved size from the user size.

(**free**): The amount of CAM currently available.

(**used**): The percentage of CAM currently being used.

Syntax: show cam-partition usage slot <*slot-number*>

Displaying CAM information

The following commands display CAM information.

Show cam l2vpn

To display all VLL or VPLS MAC entries, including local entries (Port or VLAN or MAC from end points) and remote entries (VC or MAC from VLL or VPLS peers) enter the following command.

```
NetIronr# show cam l2vpn 2/1
Slot Index    MAC            Age  Port  IFL/  VC Label  Out Port Remote DA/ PRAM
     (Hex)                          VLAN                          SA  (Hex)
2    9fff6   abcd.1234.5678  Dis  2/4   4096   74565     2/2      0     DA   8f
2    9fff7   deed.1234.5678  Dis  2/2   500    N/A       Filter   0     DA   8e
```

Syntax: show cam l2vpn <*slot/port*> [<*MAC address*>]

Show cam ipvpn

To display IPv4 VPN CAM entries, including local (Port+VLAN+IP) and remote (VC+IP) entries, enter the following command.

```
NetIron# show cam ipvpn 2/1
Slot Index   IP_Address     Port  Vlan  VC Lbl MAC            Age Out Vlan Out Port
2    0x60000 160.2.3.4/32   2/6   18    N/A    N/A            Dis 10       3/5
2    0x60001 224.7.8.9/32   N/A   N/A   4660   6336.9db8.0600 Dis 20       3/5
```

Syntax: show cam ipvpn <*slot/port*> [**IP prefix**]

Show cam label-out

To display Outbound Label ACL CAMs, enter the following command.

```
NetIron# show cam label-out 2/1
Slot Index   Port  Outer Lbl  Inner Lbl  MAC             Action
2    0xc0000 2/1   1024       1025       abcd.1234.5678  Drop
2    0xc0002 2/1   1027       1028       bade.1234.5678  Drop
```

Syntax: show cam label-out <*slot/port*>

Show IFL CAM partition

To display information about the IFL CAM partition, enter the following command.

```
NetIron#show cam ifl 2/1
Slot  Index    Port  Outer VLAN  Inner VLAN   PRAM   IFL ID
      (Hex)                                  (Hex)
2     00c5fff  2/1   100         200          185fff  4096
```

Syntax: show cam ifl <*slot/port*>

Show CAM IP

To display IP CAM information, enter the following command.

```
NetIron# show cam ip 10/1
Slot Index      IP_Address                  MAC            Age VLAN Out Port
10   0x02ff9(L) 10.10.10.10/32              N/A            Dis N/A  CPU
10   0x02ffa(L) 224.0.0.2/32                N/A            Dis N/A  CPU
10   0x02ffb(L) 224.0.0.18/32               N/A            Dis N/A  CPU
10   0x02ffc(L) 224.0.0.6/32                N/A            Dis N/A  CPU
10   0x02ffd(L) 224.0.0.5/32                N/A            Dis N/A  CPU
10   0x02ffe(L) 224.0.0.9/32                N/A            Dis N/A  CPU
10   0x02fff(L) 255.255.255.255/32          N/A            Dis N/A  CPU
10   0x3548a(R) 10.10.10.10/32              N/A            Dis N/A  Drop
10   0x3ffff(R) 0.0.0.0/0                   N/A            Dis N/A  Drop
```

Syntax: show cam ip <*slot/port*>

Show CAM IPv6

To display IPv6 CAM information, enter the following command

```
NetIron# show cam ipv6 10/3
Slot Index   IPV6_Address                    MAC            Age VLAN Out Port
10   0x5fffc fe80::0/10                      N/A            Dis N/A  CPU
10   0x5fffe ::/0                            N/A            Dis N/A  Drop
```

Syntax: show cam ipv6 <*slot/port*>

Displaying IPv6 VPN CAM information

The **show cam ipv6-vpn** command displays CAM information for an IPv6 VPN CAM entry on a single port, or for all ports on a device. IPv6 VPN CAM contains the destination IPv6 VPN address and layer 3 VPN ID. To display information for an IPv6 VPN CAM entry, enter the following command:

```
NetIron# show cam ipv6-vpn 1/1
LP Index IPV6 VPN Address                    MAC                Age
   (Hex)                                    IFL ID    Out IF   PRAM
1  407f0 10:2037::/128                        N/A                Dis
                                            (21847     Filter  1d615)
1  407f2 10:2036::/128                        N/A                Dis
                                            (21846      Drop    5af6d)
```

Syntax: show cam ipv6-vpn <*slot/port*>

Show cam v6acl

The **show cam v6acl** command displays IPv6 ACL CAM sessions configured on the device. The VLAN column is expanded to display either VLAN or IFL ID as shown in the example below:

```
NetIron# show cam v6acl 1/1
LP Index Src IP Addr                                    SPort IFL/VLAN ID
         Dst IP Addr                                    DPort Pro Age Out IF PRAM
1  74000 6000:1::/64                                    0     536977
         7000:1::/64                                    0     6   Dis Pass   000a4
1  74008 6000:2::/64                                    0     536977
         7000:2::/64                                    0     6   Dis Pass   000a5
1  74010 6000:3::/64                                    0     536977
         7000:3::/64                                    0     6   Dis Pass   000a6
1  74018 6000:4::/64                                    0     536977
         7000:4::/64                                    0     6   Dis Pass   000a7
1  74020 6000:5::/64                                    0     536977
         7000:5::/64                                    0     6   Dis Pass   000a8
1  74028 6000:6::/64                                    0     536977
```

Syntax: show cam ipv6-vpn <*slot/port*>

Show IFL CAM ISID partition

To display information about 802.1AH for ISID, enter the following command:

```
NetIron#show cam ifl-isid 1/1
Slot Index   Port   Outer VLAN Itag ISID  PRAM   IFL ID      IPV4/V6
       (Hex)                              (Hex)              Routing
1    0085fe8 1/14     27        37        185fe8  1             0/0
1    0085fe9 1/13     26        36        185fe9  1             0/1
1    0085fea 1/16     25        35        185fea  1             1/0
1    0085feb 1/15     24        34        185feb  1             1/1
```

Syntax: show cam ifl-isid <*slot/port*>

This output includes an IPv4/ IPv6 Routing column. The IPv4/IPv6 Routing column indicates whether IPv4 or IPv6 is enabled or disabled on the interface. The number 1 represents enabled, and the number 0 represents disabled. For example, if 0/0 is displayed, then IPv4/IPv6 is disabled. If 0/1 is displayed, then IPv4 is disabled/ IPv6 is enabled. The IPv4/IPv6 Routing column is also displayed in the output of the **show cam ifl** command and **show cam ifl-mpls** command.

Configuring CAM partition size

When you configure a tftp file size into the device, the device can only perform a parameter check based on the default CAM profile configured. In this situation, it is possible that you have configured a CAM partition size that conflicts with the physical CAM size. The following **system-max** commands may cause a conflict with the physical CAM size:

- system-max
  - ifl-cam
  - ip-source-guard-cam
  - ipv4-mcast-cam
  - ipv6-mcast-cam
  - lsp-out-acl-cam
  - receive cam

When you have configured a CAM size that conflicts with the physical CAM size, a partition is created with the maximum possible CAM indices assigned to it. The following Syslog message is generated:

```
NetIron# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 27 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
Sep  9 18:48:23:A:CAM IPv6 VPN SNet 9 partition warning: request 32, actual 0,
sl
ot 1, ppcr 0
```

CAM overflow logging

At system initialization, a threshold value is calculated for each sub-partition. If a partition does not have any sub-partitions, the value is based on the entire partition size. If a partition has movable sub-partition boundaries, the threshold value is also based on the entire partition size. By default, the threshold value is 5% of the total entry count. A repeat timer (default to 5 minutes) is also set for each partition to check usage. When the timer expires, if the number of unused CAM entries drops below the threshold percentage value, a log message is generated.

`CAM partition <partition name including sub-partition ID if applicable> warning: total <total count>, free <current free count>, slot <1 based slot number>, ppcr <0 based ppcr id>`

After the log message is generated, the sub-partition time stamp is updated to the current time.

Configuring minimum logging interval and threshold value

You can configure a minimum logging interval and threshold value for CAM partition logging using the following command.

```
NetIron(config)# cam-partition logging 10% 5
```

Syntax: [**no**] **cam-partition logging** <*threshold percentage*> <*interval in minutes*>

You can configure the <*threshold percentage*> variable to change the threshold value from the default 5%.

The <*interval in minutes*> variable allows you to set the minimum logging interval.

NOTE
Because IP and IPv6 sub-partitions can dynamically grow and shrink, for these partitions, logging is implemented at the entire partition level. An SNMP trap is generated with the logging message.

Appendix

Using Syslog

A

The following Syslog features are support.

- Syslog Messages
- Real-Time Display of Syslog Messages
- BFD Syslog
- Increased Syslog buffer
- Time Stamps
- Option for Show Log Command
- Disabling Logging of a Message Level
- Interface Name in Syslog Messages
- TCP or UDP Port Numbers in Syslog Messages
- Logging all CLI Commands
- BFD Logging
- Number of Entries the Local Buffer Can Hold: 1-5000

This appendix describes how to display Syslog messages and how to configure the Syslog facility, and lists the Syslog messages that a PowerConnect can display during standard operation.

NOTE
This appendix does not list Syslog messages that can be displayed when a debug option is enabled.

A PowerConnect's software can write syslog messages to provide information at the following severity levels:

- Emergencies
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

The device writes the messages to a local buffer, which can hold up to 5000 entries.

You also can specify the IP address or host name of up to six Syslog servers. When you specify a Syslog server, the PowerConnect writes the messages both to the system log and to the Syslog server.

Using a Syslog server ensures that the messages remain available even after a system reload. The PowerConnect's local Syslog buffer is cleared during a system reload or reboot, but the Syslog messages sent to the Syslog server remain on the server.

The Syslog service on a Syslog server receives logging messages from applications on the local host or from devices such as a PowerConnect. Syslog adds a time stamp to each received message and directs messages to a log file. Most Unix workstations come with Syslog configured. Some third party vendor products also provide Syslog running on NT.

Syslog uses UDP port 514 and each Syslog message thus is sent with destination port 514. Each Syslog message is one line with Syslog message format. The message is embedded in the text portion of the Syslog format. There are several subfields in the format. Keywords are used to identify each subfield, and commas are delimiters. The subfield order is insensitive except that the text subfield should be the last field in the message. All the subfields are optional.

Displaying Syslog messages

To display the Syslog messages in the device's local buffer, enter the following command at any level of the CLI.

```
NetIron# show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet 1/4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

For information about the Syslog configuration information, time stamps, and dynamic and static buffers, refer to "Displaying the Syslog configuration" on page 2255.

Enabling real-time display of Syslog messages

By default, to view Syslog messages generated by a PowerConnect, you need to display the Syslog buffer or the log on a Syslog server used by the PowerConnect.

You can enable real-time display of Syslog messages on the management console. When you enable this feature, the software displays a Syslog message on the management console when the message is generated.

When you enable the feature, the software displays Syslog messages on the serial console when they occur. However, to enable display of real-time Syslog messages in Telnet or SSH sessions, you also must enable display within the individual sessions.

To enable real-time display of Syslog messages, enter the following command at the global CONFIG level of the CLI.

```
NetIron(config)# logging console
```

Syntax: [no] logging console

This command enables the real-time display of Syslog messages on the serial console. You can enter this command from the serial console or a Telnet or SSH session.

To also enable the real-time display for a Telnet or SSH session, enter the following command from the Privileged EXEC level of the session.

```
telnet@NetIron# terminal monitor
Syslog trace was turned ON
```

Syntax: [no] terminal monitor

Notice that the CLI displays a message to indicate the status change for the feature. To disable the feature in the management session, enter the **terminal monitor** command again. The command toggles the feature on and off.

```
telnet@NetIron# terminal monitor
Syslog trace was turned OFF
```

Here is an example of how the Syslog messages are displayed.

```
telnet@NetIron# terminal monitor
Syslog trace was turned ON
SYSLOG: <9>NetIron, Power supply 2, power supply on left connector, failed

SYSLOG: <14>NetIron, Interface ethernet 1/6, state down

SYSLOG: <14>NetIron, Interface ethernet 1/2, state up
```

Configuring the Syslog service

The procedures in this section describe how to perform the following Syslog configuration tasks:

- Specify a Syslog server. You can configure the PowerConnect to use up to six Syslog servers. (Use of a Syslog server is optional. The system can hold up to 100 Syslog messages in an internal buffer.)
- Change the level of messages the system logs.
- Change the number of messages the local Syslog buffer can hold.
- Display the Syslog configuration.
- Clear the local Syslog buffer.

Logging is enabled by default, with the following settings:

- Messages of all severity levels (Emergencies – Debugging) are logged.
- By default, up to 50 messages are retained in the local Syslog buffer. This can be changed.
- No Syslog server is specified.

Displaying the Syslog configuration

To display the Syslog parameters currently in effect on a PowerConnect, enter the following command from any level of the CLI.

```
NetIron> show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet 1/4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

Syntax: show logging

The Syslog display shows the following configuration information, in the rows above the log entries themselves.

TABLE 430 CLI Display of Syslog buffer configuration

This field...	Displays...
Syslog logging	The state (enabled or disabled) of the Syslog buffer.
messages dropped	The number of Syslog messages dropped due to user-configured filters. By default, the software logs messages for all Syslog levels. You can disable individual Syslog levels, in which case the software filters out messages at those levels. refer to "Disabling logging of a message level" on page 2260. Each time the software filters out a Syslog message, this counter is incremented.
flushes	The number of times the Syslog buffer has been cleared by the **clear logging** command. refer to "Clearing the Syslog messages from the local buffer" on page 2262.
overruns	The number of times the dynamic log buffer has filled up and been cleared to hold new entries. For example, if the buffer is set for 100 entries, the 101st entry causes an overrun. After that, the 201st entry causes a second overrun.
level	The message levels that are enabled. Each letter represents a message type and is identified by the key (level code) below the value. If you disable logging of a message level, the code for that level is not listed.
messages logged	The total number of messages that have been logged since the software was loaded.
level code	The message levels represented by the one-letter codes.

Static and dynamic buffers

The software provides two separate buffers:

- **Static** – logs power supply failures, fan failures, and temperature warning or shutdown messages
- **Dynamic** – logs all other message types. In previous releases, power supply messages were displayed in static logs only, with only the last event logged in. The power supply messages are now displayed in both static and dynamic logs.

In the static log, new messages replace older ones, so only the most recent message is displayed. For example, only the most recent temperature warning message will be present in the log. If multiple temperature warning messages are sent to the log, the latest one replaces the previous one. The static buffer is not configurable.

The message types that appear in the static buffer do not appear in the dynamic buffer. The dynamic buffer contains up to the maximum number of messages configured for the buffer (50 by default), then begins removing the oldest messages (at the bottom of the log) to make room for new ones.

The static and dynamic buffers are both displayed when you display the log.

```
NetIron(config)# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
          I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dec 15 19:00:14:A:Fan 2, fan on left connector, failed

Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet 1/4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

Notice that the static buffer contains two separate messages for fan failures. Each message of each type has its own buffer. Thus, if you replace fan 1 but for some reason that fan also fails, the software replaces the first message about the failure of fan 1 with the newer message. The software does not overwrite the message for fan 2, unless the software sends a newer message for fan 2.

When you clear log entries, you can selectively clear the static or dynamic buffer, or you can clear both. For example, to clear only the dynamic buffer, enter the following command at the Privileged EXEC level.

```
NetIron# clear logging dynamic-buffer
```

Syntax: clear logging [dynamic-buffer | static-buffer]

You can specify **dynamic-buffer** to clear the dynamic buffer or **static-buffer** to clear the static buffer. If you do not specify a buffer, both buffers are cleared.

Time stamps

The contents of the time stamp differ depending on whether you have set the time and date on the onboard system clock:

- If you have set the time and date on the onboard system clock, the date and time are shown in the following format:

  mm dd hh:mm:ss

  where:

  - *mm* – abbreviation for the name of the month
  - *dd* – day

- *hh* – hours
- *mm* – minutes
- *ss* – seconds

For example, “Oct 15 17:38:03” means October 15 at 5:38 PM and 3 seconds.

- If you have not set the time and date on the onboard system clock, the time stamp shows the amount of time that has passed since the device was booted, in the following format:

  <num>d<num>h<num>m<num>s

  where:

  - <*num*>d – day
  - <*num*>h – hours
  - <*num*>m – minutes
  - <*num*>s – seconds

  For example, “188d1h01m00s” means the device had been running for 188 days, 11 hours, one minute, and zero seconds when the Syslog entry with this time stamp was generated.

Example of Syslog messages on a device whose onboard clock is set

The example shows the format of messages on a device whose onboard system clock has been set. Each time stamp shows the month, the day, and the time of the system clock when the message was generated. For example, the system time when the most recent message (the one at the top) was generated was October 15 at 5:38 PM and 3 seconds.

```
NetIron(config)# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 38 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dec 15 19:00:14:A:Fan 2, fan on left connector, failed

Dynamic Log Buffer (50 entries):
Oct 15 17:38:03:warning:list 101 denied tcp 209.157.22.191(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
Oct 15 07:03:30:warning:list 101 denied tcp 209.157.22.26(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
Oct 15 06:58:30:warning:list 101 denied tcp 209.157.22.198(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
```

Example of Syslog messages on a device whose onboard clock is not set

The example shows the format of messages on a device whose onboard system clock is not set. Each time stamp shows the amount of time the device had been running when the message was generated. For example, the most recent message, at the top of the list of messages, was generated when the device had been running for 21 days, seven hours, two minutes, and 40 seconds.

```
NetIron(config)# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 38 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:

Dynamic Log Buffer (50 entries):
21d07h02m40s:warning:list 101 denied tcp 209.157.22.191(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
19d07h03m30s:warning:list 101 denied tcp 209.157.22.26(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
17d06h58m30s:warning:list 101 denied tcp 209.157.22.198(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
```

Ascending or descending option for show log command

A new option was added to the show log command that allows you to display the log in either ascending or descending order based on time. The command will still work without the option selected and will display the log in default descending chronological order. The command is executed as shown

```
NetIron# show log ascending
```

Syntax: show log [ascending | descending]

The **ascending** option displays the oldest log entry first.

The **descending** option displays the most recent log entry first. This is the default condition.

Disabling or re-enabling Syslog

Syslog is enabled by default. To disable it, enter the following command at the global CONFIG level.

```
NetIron(config)# no logging on
```

Syntax: [no] logging on [<*udp-port*>**]**

The <*udp-port*> parameter specifies the application port used for the Syslog facility. The default is 514.

To re-enable logging, enter the following command.

```
NetIron(config)# logging on
```

This command enables local Syslog logging with the following defaults:

- Messages of all severity levels (Emergencies – Debugging) are logged.
- Up to 50 messages are retained in the local Syslog buffer.
- No Syslog server is specified.

Specifying a Syslog server

To specify a Syslog server, enter a command such as the following

```
NetIron(config)# logging host 10.0.0.99
```

For backward compatibility, the software reads the old command syntax from the startup configuration, and converts it to the new command syntax in the running configuration.

Syntax: [no] logging host <*ip-addr*> | <*server-name*>

Specifying an additional Syslog server

To specify an additional Syslog server, enter the **logging host** <*ip-addr*> command again, as in the following example. You can specify up to six Syslog servers.

Enter a command such as the following

```
NetIron(config)# logging host 10.0.0.99
```

For backward compatibility, the software reads the old command syntax from the startup configuration, and converts it to the new command syntax in the running configuration.

Syntax: [no] logging host <*ip-addr*> | <*server-name*>

Disabling logging of a message level

If you want to disable the logging of a message level, you must disable each message level individually.

For example, to disable logging of debugging and informational messages, enter the following commands

```
NetIron(config)# no logging buffered debugging
NetIron(config)# no logging buffered informational
```

Syntax: [no] logging buffered <*level*> | <*num-entries*>

The <*level*> parameter can have one of the following values:

- alerts
- critical
- debugging
- emergencies
- errors
- informational
- notifications
- warnings

The commands in the example above change the log level to notification messages or higher. The software will not log informational or debugging messages. The changed message level also applies to the Syslog servers.

Changing the number of entries for the local buffer

You also can use the **logging buffered** command to change the number of entries the local Syslog buffer can store.

Example

```
NetIron(config)# logging buffered 100
```

Syntax: [no] logging buffered <*level*> | <*num-entries*>

The default number of messages is 50. The change takes effect immediately and does not require you to reload the software.

Changing the log facility

The Syslog daemon on the Syslog server uses a facility to determine where to log the messages from the PowerConnect. The default facility for messages the PowerConnect sends to the Syslog server is "user". You can change the facility using the following command.

NOTE
You can specify only one facility. If you configure the PowerConnect to use two Syslog servers, the device uses the same facility on both servers.

```
NetIron(config)# logging facility local0
```

Syntax: [no] logging facility <*facility-name*>

The <*facility-name*> can be one of the following:

- kern – kernel messages
- user – random user-level messages
- mail – mail system
- daemon – system daemons
- auth – security or authorization messages
- syslog – messages generated internally by Syslog
- lpr – line printer subsystem
- news – netnews subsystem
- uucp – uucp subsystem
- sys9 – cron or at subsystem
- sys10 – reserved for system use
- sys11 – reserved for system use
- sys12 – reserved for system use
- sys13 – reserved for system use
- sys14 – reserved for system use
- cron – cron or at subsystem
- local0 – reserved for local use
- local1 – reserved for local use
- local2 – reserved for local use
- local3 – reserved for local use
- local4 – reserved for local use
- local5 – reserved for local use
- local6 – reserved for local use
- local7 – reserved for local use

Displaying the interface name in Syslog messages

By default, an interface's slot number (if applicable) and port number are displayed when you display Syslog messages. If you want to display the name of the interface instead of its number, enter the following command.

```
NetIron(config)# ip show-portname
```

This command is applied globally to all interfaces on the PowerConnect.

Syntax: [no] Ip show-portname

When you display the messages in the Syslog, you refer to the interface name under the Dynamic Log Buffer section. The actual interface number is appended to the interface name. For example, if the interface name is "lab" and its port number is "2", you refer to "lab2" displayed as in the example below.

```
NetIron# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
         I=informational N=notification W=warning
Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet Lab2, state up
Dec 15 18:45:15:I:Warm start
```

Clearing the Syslog messages from the local buffer

To clear the Syslog messages stored in the PowerConnect's local buffer, use the following command.

```
NetIron# clear logging
```

Syntax: clear logging

Displaying TCP or UDP port numbers in Syslog messages

The command **ip show-acl-service-number** allows you to change the display of TCP or UDP application information from the TCP or UDP well-known port name to the TCP or UDP port number. For example, entering the following command causes the PowerConnect to display http (the well-known port name) instead of 80 (the port number) in the output of show commands, and other commands that contain application port information. By default, the PowerConnect displays TCP or UDP application information in named notation.

In this release, you can display TCP or UDP port number instead of their names in syslog messages by entering the following command.

```
NetIron(config)# ip show-service-number-in-log
```

Syntax: [no] ip show-service-number-in-log

Logging all CLI commands to Syslog

This feature allows you to log all valid CLI command from each user session into the system log.

To enable CLI command logging, enter the following command.

```
NetIron(config)# logging cli-command
```

Syntax: [no] logging cli-command

Example of CLI command logging

In the following example, two CLI sessions are run. In the first example, a telnet session enables CLI command logging and configures **router bgp** and the BGP **no neighbor** command as shown.

```
telnet@ NetIron-2(config)# logging cli-command
telnet@ NetIron-2(config)# router bgp
telnet@ NetIron-2(config-bgp)# no nei 10.1.1.8 remote 10
```

In the next example, a console session configures **router bgp** and the BGP **neighbor** command as shown.

```
NetIron-2(config)# router bgp
NetIron-2(config-bgp)# nei 10.1.1.8 remote 10
```

Using the **show log** command, you would refer to a series of log records as shown in the following.

```
NetIron(config-bgp)# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 24 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
Dynamic Log Buffer (50 lines):
Sep 9 18:38:23:I:CLI CMD: "nei 10.1.1.8 remote 10" from console
Sep 9 18:38:21:I:CLI CMD: "router bgp" from console
Sep 9 18:38:07:I:CLI CMD: "no nei 10.1.1.8 remote 10" from telnet client 10.1.1.1
Sep 9 18:38:05:I:CLI CMD: "router bgp" from telnet client 10.1.1.1
```

Syslog messages

The tables that follow list all of the Syslog messages. The messages are listed by message level, in the following order:

- Emergencies (none)
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

TABLE 431 Syslog messages system

Message level	Message	Explanation
Alert	ISIS Memory Limit Exceeded	IS-IS is requesting more memory than is available.
Alert	System Power supply <*num*>, <*location*>, failed	A power supply has failed. The <*num*> is the power supply number. The <*location*> describes where the failed power supply is in the chassis.
Alert	System <*power type*> Power Supply <*num*>, <*location*>,< *state*>	The <*power type*> refers to AC or DC power supply. The <*num*> is the power supply number as positioned in the chassis. The <*location*> describes where the power supply is in the chassis in relation to its state. The <*state*> refers to how the power supply is functioning in the chassis. The <*state*> can be one of the following: • Installed (OK) - The power supply is installed and ok. • Installed (Failed or Disconnected)- The power supply has failed, or the power cord is disconnected. • Not Installed (FAILED) - The power supply is physically removed from the chassis.
Alert	System Fan <*num*>, <*location*>, failed	A fan has failed. The <*num*> is the power supply number. The <*location*> describes where the failed power supply is in the chassis. The location can be one of the following
Alert	System Management module at slot <*slot-num*> state changed from <*module-state*> to <*module-state*> due to <*reason*>.	Indicates a state change in a management module. The <*slot-num*> indicates the chassis slot containing the module. The <*module-state*> can be one of the following: • active • standby • crashed • coming-up • unknown A **due to** clause has been added to this message. The <*reason*> variable can be either or the following: • MP upgrade to ver <*version number*> where <*version number*> is the version number of the Multi-Service IronWare software that the management module was upgraded to. • Active Reboot
Alert	System Temperature <*degrees*> C degrees, warning level <*warn-degrees*> C degrees, shutdown level <*shutdown-degrees*> C degrees	Indicates an overtemperature condition on the active module. The <*degrees*> value indicates the temperature of the module. The <*warn-degrees*> value is the warning threshold temperature configured for the module. The <*shutdown-degrees*> value is the shutdown temperature configured for the module.

TABLE 431 Syslog messages system (Continued)

Message level	Message	Explanation
Alert	System <*num-modules*> modules and 1 power supply, need more power supply!!	Indicates that the chassis needs more power supplies to run the modules in the chassis. The <*num-modules*> parameter indicates the number of modules in the chassis.
Alert	System: Health Monitoring: TM Egress data errors detected on LP <*num*>/TM <*num*>	The system has detected egress data errors on the specified line processor and traffic manager. The LP and TM <*num*> parameters indicate the number of the line processor and traffic manager on which the errors were detected.
Error	Error Module down in slot 3, reason CARD_DOWN_REASON_BOOT_FAILED.Err or Code (1).	• The error message displayed on the Management Module console when the Interface Module fails to boot up. The message will display the error code reason. • When the Interface Module is in DOWN state, the error code is included in the dynamic buffer. The error code is 0 when there is no error code reported from the Interface Module.
Warning	CAM partition <*partition name*> warning total <*total-count*>, free <current free-count>, slot <*slot-number*>, ppcr <*ppcr-id*>	Indicates that the CAM partition specified by the <*partition name*> has exceeded a threshold (configurable with a default value of within 5% of the capacity of the partition) and may soon overflow the threshold. The <*free-count*> specifies the amount of free space still available in the partition. The <*slot-number*> and ppcr <*ppcr-id*> indicate where the overflow is occurring. The <*partition-name*> includes the sub-partition ID if applicable.
Warning	System Not enough power to power on module in slot <*num*>	There is not enough power available in the chassis to power on the module in the specific slot number. The slot <*num*> refers to the slot number in the chassis.
Notification	System Module up in slot <*n*>	
Notification	System Module down in slot <*n*> reason <*reason*>	Indicates that the module in the slot specified by the <*n*> variable is down for one of the following reasons as specified by the <*reason*> variable: • CARD_DOWN_REASON_NONE • CARD_DOWN_REASON_ADMIN_DOWN • CARD_DOWN_REASON_CONFIG_MISMATCH • CARD_DOWN_REASON_LOSS_HEARTBEAT • CARD_DOWN_REASON_BOOT_FAILED • CARD_DOWN_REASON_TIMEOUT • CARD_DOWN_REASON_STRIPE_SYNC_FAILED • CARD_DOWN_REASON_REBOOTED • CARD_DOWN_REASON_OVER_HEAT • CARD_DOWN_REASON_POWERED_OFF_BY_USER • CARD_DOWN_REASON_LINK_DOWN

TABLE 431 Syslog messages system (Continued)

Message level	Message	Explanation
Notification	System Module <*n*> powered on	
Notification	System Module <*n*> powered off	
Notification	System Switch fabric <*n*> powered on	
Notification	System Switch fabric <*n*> powered off	
Notification	System Enough power available to power on module in slot <*num*>.	There is enough power available in the chassis to power on the module in the specific slot number. The slot <*num*> refers to the slot number in the chassis.
Notification	System Module was inserted to slot <*slot-num*>	Indicates that a module was inserted into a chassis slot. The <*slot-num*> is the number of the chassis slot into which the module was inserted.
Notification	System Module was removed from slot <*slot-num*>	Indicates that a module was removed from a chassis slot. The <*slot-num*> is the number of the chassis slot from which the module was removed.
Notification	System Set fan speed to <*speed*> <*percentage*>	Indicates that the fan speed has been changed to the value described in the <*speed*> variable and that the fan is now operating at the <*percentage*> of capacity described. The possible <*speed*> <*percentage*> values are: • LOW (50%) • MEDIUM (75%) • MEDIUM-HIGH (90%) • HIGH (100%)
Informational	System Cold start	The device has been powered on.
Informational	System Warm start	The system software (flash code) has been reloaded.
Informational	System Interface <*portnum*>, state up	A port has come up. The <*portnum*> is the port number.
Informational	System Interface <*portnum*>, state down	A port has gone down. The <*portnum*> is the port number.
Informational	System Interface <*portnum*>, line protocol up	The line protocol on a port has come up. The <*portnum*> is the port number.
Informational	System Interface <*portnum*>, line protocol down	The line protocol on a port has gone down. The <*portnum*> is the port number.
Informational	System Interface <*portnum*> is down (remote fault)	The interface is down due to Remote Fault. This is indicated as “(remote fault)”. The <*portnum*> is the port number of the interface.
Informational	System <*portnum*> is down (local fault)	The port is down due to Local Fault. This is indicated as “(local fault)”. The <*portnum*> is the port number of the interface.

TABLE 431 Syslog messages system (Continued)

Message level	Message	Explanation
Informational	System Syslog server <*IP-address*> deleted | added | modified from console | telnet | ssh | web | snmp OR Syslog operation enabled | disabled from console | telnet | ssh | web | snmp	A user made Syslog configuration changes to the specified Syslog server address, or enabled or disabled a Syslog operation through the Web, SNMP, console, SSH, or Telnet session.
Informational	System SSH | telnet server enabled | disabled from console | telnet | ssh | web | snmp session [by user <*username*>]	A user enabled or disabled an SSH or Telnet session, or changed the SSH enable or disable configuration through the Web, SNMP, console, SSH, or Telnet session.
Informational	System Module <*n*> CPU <*m*> crashed	
Informational	System IfIndex assignment was changed .)	The maximum number of ifIndex per module has been changed.

TABLE 432 Syslog messages security

Message level	Message	Explanation
Warning	Security Port security violation at interface e<*portnum*>, address <*mac*>, vlan <*id*>	
Warning	Security Interface e<*portnum*> was shut down due to port security violation	
Informational	Security console login {by <*user*>I<*null*>} to USER EXEC mode Security {telnet I ssh} login {by <*user*>I<*null*>} from src {IP <*ip*> I IPv6 <*ipv6-addr*>} to USER EXEC mode	A user has logged into the USER EXEC mode of the CLI. The <*user*> is the user name.
Informational	Security console logout {by <*user*>I<*null*>} from USER EXEC mode Security {telnet I ssh} logout {by <*user*>I<*null*>} from src {IP <*ip*> I IPv6 <*ipv6-addr*>} from USER EXEC mode	A user has logged out of the USER EXEC mode of the CLI. The <*user*> is the user name.
Informational	Security console login {by <*user*>I<*null*>} to Privileged EXEC mode Security {telnet I ssh} login {by <*user*>I<*null*>} from src {IP <*ip*> I IPv6 <*ipv6-addr*>} to Privileged EXEC mode	A user has logged into the Privileged EXEC mode of the CLI. The <*user*> is the user name.
Informational	Security console logout {by <*user*>I<*null*>} from Privileged EXEC mode Security {telnet I ssh} logout {by <*user*>I<*null*>} from src {IP <*ip*> I IPv6 <*ipv6-addr*>} from Privileged EXEC mode	A user has logged out of Privileged EXEC mode of the CLI. The <*user*> is the user name.
Informational	Security outbound telnet <*session number*> login to server IP <*ip*> from SSH session <*session number*>	A user has initiated an outbound Telnet session from an inbound SSH session. The first <*session number*> is the number of the outbound Telnet session. The <*ip*> is the IP address to which the Telnet session is connected. The second <*sessions number*> is the number of the inbound SSH session.

TABLE 432 Syslog messages security (Continued)

Message level	Message	Explanation
Informational	Security outbound telnet <*session number*> logout from server IP <*ip*> from SSH session <*session number*>	A user has terminated an outbound Telnet session initiated from an inbound SSH session. The first <*session number*> is the number of the outbound Telnet session. The <*ip*> is the IP address from which the Telnet session has disconnected. The second <*sessions number*> is the number of the inbound SSH session.
Informational	Security startup-config was changed {by <*user*>} from {web management I snmp management I ssh client <*ip*> I telnet client <*ip*>}	A configuration change was saved to the startup configuration file. The <*user*> is the user's ID, if they entered a user ID to log in.
Informational	Security running-config was changed {by <*user*>} from {web management I snmp management I ssh client <*ip*> I telnet client <*ip*>}	A configuration change was saved to the running configuration file. The <*user*> is the user's ID, if they entered a user ID to log in.
Informational	Security telnet | SSH | web access [by <*username*>] from src IP <*source ip address*>, src MAC <*source MAC address*> rejected, <*n*> attempts	There were failed web, SSH, or Telnet login access attempts from the specified source IP and MAC address. • [by <*user*> <*username*>] does not appear if telnet or SSH clients are specified. • <*n*> is the number of times this SNMP trap occurred in the last five minutes, or other configured number of minutes.
Informational	Security user <*username*> added | deleted | modified from console | telnet | ssh | web | snmp	A user created, modified, or deleted a local user account through the Web, SNMP, console, SSH, or Telnet session.
Informational	Security Enable super | port-config | read-only password deleted | added | modified from console | telnet | ssh | web | snmp OR Line password deleted | added | modified from console | telnet | ssh | web | snmp	A user created, re-configured, or deleted an Enable or Line password through the Web, SNMP, console, SSH, or Telnet session.

TABLE 433 Syslog messages VLAN

Message level	Message	Explanation
Informational	VLAN Id <*vlan-id*> added | deleted | modified from console | telnet | ssh | web | snmp session	A user created, modified, or deleted a VLAN through the Web, SNMP, console, SSH, or Telnet session.

TABLE 434 Syslog messages STP

Message level	Message	Explanation
Informational	STP VLAN <*id*> - New RootBridge <*string*> RootPort <*portnum*> (<*reason*>)	A Spanning Tree Protocol (STP) topology change has occurred. The <*id*> is the ID of the VLAN in which the STP topology change occurred. The <*portnum*> is the number of the port connected to the new root bridge.
Informational	STP VLAN <*id*> - Bridge is RootBridge <*string*> (<*reason*>)	A Spanning Tree Protocol (STP) topology change has occurred, resulting in the PowerConnect becoming the root bridge. The <*id*> is the ID of the VLAN in which the STP topology change occurred.
Informational	STP VLAN <*id*> Port <*portnum*> - Bridge TC Event (<*reason*>)	A Spanning Tree Protocol (STP) topology change has occurred on a port. The <*id*> is the ID of the VLAN in which the STP topology change occurred. The <*portnum*> is the port number.
Informational	STP VLAN <*vlanid*> Port <*portnum*> - State <*state*> (<*reason*>)	
Informational	STP Root Guard Port <*portnum*>, VLAN <*vlan-id*> inconsistent (Received superior BPDU)	The specified port was blocked because i t has Root Guard enabled and received a superior BPDU .
Informational	STP Root Guard Port <*portnum*>, VLAN <*vlan-id*> consistent (Timeout)	The specified block Root Guard-protected port was unblocked.
Informational	STP BPDU Guard port <*portnum*> disable System Interface ethernet <*portnum*>, state down - disabled	The **spanning-tree protect do-disable** command is configured on the specified port and the port became disabled due to a receipt of a BPDU packet.
Informational	STP BPDU Guard re-enabled on ports ethe <*portnum*> System Interface ethernet <*portnum*>, state up	The **spanning-tree protect re-enable** was issued to re-enable the specified port

TABLE 435 Syslog messages RSTP

Message level	Message	Explanation
Informational	RSTP VLAN <*id*> Port <*portnum*> - Bridge TC Event (reason)	802.1W recognized a topology change event in the bridge. The topology change event is the forwarding action that started on a non-edge Designated port or Root port.
Informational	RSTP VLAN <*id*> Port <*portnum*> - STP State <*state*> (reason)	802.1W changed the state of a port to a new state forwarding, learning, blocking. If the port changes to blocking, the bridge port is in discarding state.
Informational	RSTP VLAN <*id*> - New RootPort <*portnum*> (reason)	802.1W changed the port's role to Root port, using the root selection computation.
Informational	RSTP VLAN <*id*> - New RootBridge <*string*> RootPort <*portnum*> (reason)	802.1W selected a new root bridge as a result of the BPDUs received on a bridge port.

TABLE 435 Syslog messages RSTP (Continued)

Message level	Message	Explanation
Informational	RSTP VLAN <*id*> - Bridge is RootBridge <*string*> (reason)	802.1W changed the current bridge to be the root bridge of the given topology due to administrative change in bridge priority.
Informational	vlan <*vlan-id*> Bridge is RootBridge <*mac-address*> (MsgAgeExpiry)	The message age expired on the Root port so 802.1W changed the current bridge to be the root bridge of the topology.

TABLE 436 Syslog messages LAG

Message level	Message	Explanation
Informational	LAG group (<*ports*>) created by 802.3ad link-aggregation module.	802.3ad link aggregation is configured on the device, and the feature has dynamically created a LAG group (aggregate link). The <*ports*> is a list of the ports that were aggregated to make the LAG group.

TABLE 437 Syslog messages MRP

Message level	Message	Explanation
Informational	MRP interface ethernet <*portnum*> vlan <*vlan-master*>, changing to <*state-string*>	
Informational	MRP metro ring <*ring-id*> cannot be enabled. No free CAM entries	

TABLE 438 Syslog messages UDLD

Message level	Message	Explanation
Informational	UDLD Logical link on interface ethernet <*portnum*> is up	
Informational	UDLD Logical link on interface ethernet <*portnum*> is down	

TABLE 439 Syslog messages VSRP

Message level	Message	Explanation
Informational	VSRP VLAN <*vlanid*> VRID <*id*> - transition to <*state-string*>	
Informational	VSRP VLAN <*vlanid*> VRID <*id*> - aware change <*old-portnum*> -> <*new-portnum*>\n	
Informational	VSRP VLAN <*vlanid*> VRID <*id*> - aware learn <*portnum*>	

TABLE 440 Syslog messages VRRP

Message level	Message	Explanation
Notification	VRRP intf state changed, intf <*portnum*>, vrid <*virtual-router-id*>, state <*vrrp-state*>	A state change has occurred in a Virtual Router Redundancy Protocol (VRRP) interface. The <*portnum*> is the port. The <*virtual-router-id*> is the virtual router ID (VRID) configured on the interface. The <*vrrp-state*> can be one of the following: • init • master • backup • unknown

TABLE 441 Syslog messages IP

Message level	Message	Explanation
Warning	IP Dup IP <*ip-addr*> detected, sent from MAC <*mac-addr*> interface <*portnum*>	Indicates that the PowerConnect received a packet from another device on the network with an IP address that is also configured on the PowerConnect. The <*ip-addr*> is the duplicate IP address. The <*mac-addr*> is the MAC address of the device with the duplicate IP address. The <*portnum*> is the port that received the packet with the duplicate IP address. The address is the packet's source IP address.

TABLE 442 Syslog messages ICMP

Message level	Message	Explanation
Notification	ICMP Local ICMP exceeds <*burst-max*> burst packets, stopping for <*lockup*> seconds!!	The number of ICMP packets exceeds the <*burst-max*> threshold set by the **ip icmp burst** command. The PowerConnect may be the victim of a Denial of Service (DoS) attack. All ICMP packets will be dropped for the number of seconds specified by the <*lockup*> value. When the lockup period expires, the packet counter is reset and measurement is restarted.
Notification	ICMP Transit ICMP in interface <*portnum*> exceeds <*num*> burst packets, stopping for <*num*> seconds!!	Threshold parameters for ICMP transit (through) traffic have been configured on an interface, and the maximum burst size for ICMP packets on the interface has been exceeded. The <*portnum*> is the port number. The first <*num*> is the maximum burst size (maximum number of packets allowed). The second <*num*> is the number of seconds during which additional ICMP packets will be blocked on the interface. **NOTE:** This message can occur in response to an attempted Smurf attack.

TABLE 443 Syslog messages ACL

Message level	Message	Explanation
Warning	ACL list <*acl-num*> denied <*ip-proto*> <*src-ip-addr*> (<*src-tcp/udp-port*>) (Ethernet <*portnum*> <*mac-addr*>) -> <*dst-ip-addr*> (<*dst-tcp/udp-port*>), 1 events	Indicates that an Access Control List (ACL) denied (dropped) packets. The <*acl-num*> indicates the ACL number. Numbers 1 – 99 indicate standard ACLs. Numbers 100 – 199 indicate extended ACLs. The <*ip-proto*> indicates the IP protocol of the denied packets. The <*src-ip-addr*> is the source IP address of the denied packets. The <*src-tcp/udp-port*> is the source TCP or UDP port, if applicable, of the denied packets. The <*portnum*> indicates the port number on which the packet was denied. The <*mac-addr*> indicates the source MAC address of the denied packets. The <*dst-ip-addr*> indicates the destination IP address of the denied packets. The <*dst-tcp/udp-port*> indicates the destination TCP or UDP port number, if applicable, of the denied packets.
Warning	ACL:rip filter list <*list-num*> <*direction*> V1 | V2 denied <*ip-addr*>, <*num*> packets	Indicates that a RIP route filter denied (dropped) packets. The <*list-num*> is the ID of the filter list. The <*direction*> indicates whether the filter was applied to incoming packets or outgoing packets. The value can be one of the following: • in • out The V1 or V2 value specifies the RIP version (RIPv1 or RIPv2). The <*ip-addr*> indicates the network number in the denied updates. The <*num*> indicates how many packets matching the values above were dropped during the five-minute interval represented by the log entry.
Notification	ACL insufficient L4 session resource, using flow based ACL instead	The device does not have enough Layer 4 session entries. To correct this condition, allocate more memory for sessions. To allocate more memory, enter the following command at the global CONFIG level of the CLI interface **system-max session-limit** <*num*>
Notification	ACL system fragment packet inspect rate <*rate*> exceeded	The fragment rate allowed on the device has been exceeded. The <*rate*> indicates the maximum rate allowed. This message can occur if fragment thottling is enabled.

TABLE 443 Syslog messages ACL (Continued)

Message level	Message	Explanation
Notification	AC port fragment packet inspect rate <*rate*> exceeded on port <*portnum*>	The fragment rate allowed on an individual interface has been exceeded. The <*rate*> indicates the maximum rate allowed. The <*portnum*> indicates the port. This message can occur if fragment thottling is enabled.
Notification	ACL Port <*portnum*>, exceed configured L4 rule-based CAM size, larger L4 partition size required	
Notification	ACL Port <*portnum*>, exceed configured L2 ACL rule-based CAM size, larger partition size is required	
Notification	ACL Port <*portnum*>, exceed configured outbound L4 rule-based CAM size, larger outbound L4 partition size required	
Notification	ACL Port <*portnum*>, exceed configured IPv6 L4 rule-based CAM size, larger IPv6 L4 partition size required	
Notification	ACL Port <*portnum*>, exceed configured IPv6 outbound L4 rule-based CAM size, larger IPv6 outbound L4 partition size required	
Notification	ACL Port <*portnum*>, error in allocating inbound L4 rule-based ACL CAM entry	
Notification	ACL Port <*portnum*>, error in allocating outbound L4 rule-based ACL CAM entry	
Notification	ACL Port <*portnum*>, inbound ACL CAM programming incomplete	
Notification	ACL Port <*portnum*>, outbound ACL CAM programming incomplete	
Informational	ACL <*aclid*> added | deleted | modified from console | telnet | ssh | web | snmp session	A user created, modified, deleted, or applied an ACL through the Web, SNMP, console, SSH, or Telnet session.

TABLE 444 Syslog messages RACL

Message level	Message	Explanation
Notification	RACL Port <*portnum*>, IP Receive ACL exceed configured CAM size, larger partition size required	
Notification	RACL Port <*portnum*>, IP Receive ACL exceed configured RL class limit	
Notification	RACL Port <*portnum*>, IP Receive ACL CAM malloc error	

TABLE 445 Syslog messages OSPF

Message level	Message	Explanation
Alert	OSPF Memory Overflow	OSPF has run out of memory.
Alert	OSPF LSA Overflow, LSA Type = <*lsa-type*>	Indicates an LSA database overflow. The <*lsa-type*> parameter indicates the type of LSA that experienced the overflow condition. The LSA type is one of the following: • 1 – Router • 2 – Network • 3 – Summary • 4 – Summary • 5 – External
Notification	OSPF interface state changed, rid <*router-id*>, intf addr <*ip-addr*>, state <*ospf-state*>	Indicates that the state of an OSPF interface has changed. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the interface's IP address. The <*ospf-state*> indicates the state to which the interface has changed and can be one of the following: • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown
Notification	OSPF virtual intf state changed, rid <*router-id*>, area <*area-id*>, nbr <*ip-addr*>, state <*ospf-state*>	Indicates that the state of an OSPF virtual routing interface has changed. The <*router-id*> is the router ID of the router the interface is on. The <*area-id*> is the area the interface is in. The <*ip-addr*> is the IP address of the OSPF neighbor. The <*ospf-state*> indicates the state to which the interface has changed and can be one of the following: • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF nbr state changed, rid <*router-id*>, nbr addr <*ip-addr*>, nbr rid <*nbr-router-ld*>, state <*ospf-state*>	Indicates that the state of an OSPF neighbor has changed. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the neighbor. The <*nbr-router-id*> is the router ID of the neighbor. The <*ospf-state*> indicates the state to which the interface has changed and can be one of the following: • down • attempt • initializing • 2-way • exchange start • exchange • loading • full • unknown
Notification	OSPF virtual nbr state changed, rid <*router-id*>, nbr addr <*ip-addr*>, nbr rid <*nbr-router-id*>, state <*ospf-state*>	Indicates that the state of an OSPF virtual neighbor has changed. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the neighbor. The <*nbr-router-id*> is the router ID of the neighbor. The <*ospf-state*> indicates the state to which the interface has changed and can be one of the following: • down • attempt • initializing • 2-way • exchange start • exchange • loading • full • unknown

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF intf config error, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, error type <*error-type*>, pkt type <*pkt-type*>	Indicates that an OSPF interface configuration error has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the error packet. The <*error-type*> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF virtual intf config error, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, error type <*error-type*>, pkt type <*pkt-type*>	Indicates that an OSPF virtual routing interface configuration error has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the error packet. The <*error-type*> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF intf authen failure, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, error type <*error-type*>, pkt type <*pkt-type*>	Indicates that an OSPF interface authentication failure has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the authentication failure. The <*error-type*> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF virtual intf authen failure, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, error type <*error-type*>, pkt type <*pkt-type*>	Indicates that an OSPF virtual routing interface authentication failure has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the authentication failure. The <*error-type*> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF intf rcvd bad pkt, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, pkt type <*pkt-type*>	Indicates that an OSPF interface received a bad packet. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the authentication failure. The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown **NOTE:** This message is typically generated during BFD or OSPF reconverge within the following scenarios: • The router is undergoing hitless upgrade • Management module switchover, • Interface module CPU utilization is at 95% or more, • The **clear ip ospf neighbor all** command is issued. During these processes, OSPF adj is deleted due to BFD time out while the router can still receive OSPF packets destined to a previous session from its neighbor because the neighbor has an inconsistent OSPF state due to timing. This message will go away shortly when BFD or OSPF re-establishes neighbor.
Notification	OSPF virtual intf rcvd bad pkt, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, pkt type <*pkt-type*>	Indicates that an OSPF interface received a bad packet. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the authentication failure. The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF intf retransmit, rid <*router-id*>, intf addr <*ip-addr*>, nbr rid <*nbr-router-id*>, pkt type is <*pkt-type*>, LSA type <*lsa-type*>, LSA id <*lsa-id*>, LSA rid <*lsa-router-id*>	An OSPF interface on the PowerConnect has retransmitted a Link State Advertisement (LSA). The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*nbr-router-id*> is the router ID of the neighbor router. The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown The <*lsa-type*> is the type of LSA. The <*lsa-id*> is the LSA ID. The <*lsa-router-id*> is the LSA router ID.
Notification	OSPF virtual intf retransmit, rid <*router-id*>, intf addr <*ip-addr*>, nbr rid <*nbr-router-id*>, pkt type is <*pkt-type*>, LSA type <*lsa-type*>, LSA id <*lsa-id*>, LSA rid <*lsa-router-id*>	An OSPF interface on the PowerConnect has retransmitted a Link State Advertisement (LSA). The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*nbr-router-id*> is the router ID of the neighbor router. The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown The <*lsa-type*> is the type of LSA. The <*lsa-id*> is the LSA ID. The <*lsa-router-id*> is the LSA router ID.
Notification	OSPF originate LSA, rid <*router-id*>, area <*area-id*>, LSA type <*lsa-type*>, LSA id <*lsa-id*>, LSA router id <*lsa-router-id*>	An OSPF interface has originated an LSA. The <*router-id*> is the router ID of the PowerConnect. The <*area-id*> is the OSPF area. The <*lsa-type*> is the type of LSA. The <*lsa-id*> is the LSA ID. The <*lsa-router-id*> is the LSA router ID.

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF max age LSA, rid <*router-id*>, area <*area-id*>, LSA type <*lsa-type*>, LSA id <*lsa-id*>, LSA rid <*lsa-router-id*>	An LSA has reached its maximum age. The <*router-id*> is the router ID of the PowerConnect. The <*area-id*> is the OSPF area. The <*lsa-type*> is the type of LSA. The <*lsa-id*> is the LSA ID. The <*lsa-router-id*> is the LSA router ID.
Notification	OSPF LSDB overflow, rid <*router-id*>, limit <*num*>	A Link State Database Overflow (LSDB) condition has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*num*> is the number of LSAs.
Notification	OSPF LSDB approaching overflow, rid <*router-id*>, limit <*num*>	The software is close to an LSDB condition. The <*router-id*> is the router ID of the PowerConnect. The <*num*> is the number of LSAs.
Notification	OSPF intf rcvd bad pkt Bad Checksum, rid <*ip-addr*>, intf addr <*ip-addr*>, pkt size <*num*>, checksum <*num*>, pkt src addr <*ip-addr*>, pkt type <*type*>	The device received an OSPF packet that had an invalid checksum. The rid <*ip-addr*> is PowerConnect's router ID. The intf addr <*ip-addr*> is the IP address of the interface that received the packet. The pkt size <*num*> is the number of bytes in the packet. The checksum <*num*> is the checksum value for the packet. The pkt src addr <*ip-addr*> is the IP address of the neighbor that sent the packet. The pkt type <*type*> is the OSPF packet type and can be one of the following: • hello • database description • link state request • link state update • link state acknowledgement • unknown (indicates an invalid packet type)
Notification	OSPF intf rcvd bad pkt Bad Packet type, rid <*ip-addr*>, intf addr <*ip-addr*>, pkt size <*num*>, checksum <*num*>, pkt src addr <*ip-addr*>, pkt type <*type*>	The device received an OSPF packet with an invalid type. The parameters are the same as for the Bad Checksum message. The pkt type <*type*> value is "unknown", indicating that the packet type is invalid.

TABLE 445 Syslog messages OSPF (Continued)

Message level	Message	Explanation
Notification	OSPF intf rcvd bad pkt Unable to find associated neighbor, rid <*ip-addr*>, intf addr <*ip-addr*>, pkt size <*num*>, checksum <*num*>, pkt src addr <*ip-addr*>, pkt type <*type*>	The neighbor IP address in the packet is not on the PowerConnect's list of OSPF neighbors. The parameters are the same as for the Bad Checksum message.
Notification	OSPF intf rcvd bad pkt Invalid packet size, rid <*ip-addr*>, intf addr <*ip-addr*>, pkt size <*num*>, checksum <*num*>, pkt src addr <*ip-addr*>, pkt type <*type*>	The device received an OSPF packet with an invalid packet size. The parameters are the same as for the Bad Checksum message.

TABLE 446 Syslog messages OSPFv3

Message level	Message	Explanation
Alert	OSPFv3 Memory Overflow	OSPF has run out of memory.
Alert	OSPFv3 LSA Overflow, LSA Type = <*lsa-type*>	Indicates an LSA database overflow. The <*lsa-type*> parameter indicates the type of LSA that experienced the overflow condition. The LSA type is one of the following: • 1 – Router • 2 – Network • 3 – Summary • 4 – Summary • 5 – External
Notification	OSPFv3 interface state changed, rid <*router-id*>, intf addr <*ip-addr*>, state <*ospf-state*>	Indicates that the state of an OSPF interface has changed. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the interface's IP address. The <*ospf-state*> indicates the state to which the interface has changed and can be one of the following: • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown

TABLE 446 Syslog messages OSPFv3 (Continued)

Message level	Message	Explanation
Notification	OSPFv3 virtual intf state changed, rid <*router-id*>, area <*area-id*>, nbr <*ip-addr*>, state <*ospf-state*>	Indicates that the state of an OSPF virtual routing interface has changed. The <*router-id*> is the router ID of the router the interface is on. The <*area-id*> is the area the interface is in. The <*ip-addr*> is the IP address of the OSPF neighbor. The <*ospf-state*> indicates the state to which the interface has changed and can be one of the following: • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown
Notification	OSPFv3 nbr state changed, rid <*router-id*>, nbr addr <*ip-addr*>, nbr rid <*nbr-router-ld*>, state <*ospf-state*>	Indicates that the state of an OSPF neighbor has changed. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the neighbor. The <*nbr-router-id*> is the router ID of the neighbor. The <*ospf-state*> indicates the state to which the interface has changed and can be one of the following: • down • attempt • initializing • 2-way • exchange start • exchange • loading • full • unknown
Notification	OSPFv3 virtual nbr state changed, rid <*router-id*>, nbr addr <*ip-addr*>, nbr rid <*nbr-router-id*>, state <*ospf-state*>	Indicates that the state of an OSPF virtual neighbor has changed. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the neighbor. The <*nbr-router-id*> is the router ID of the neighbor. The <*ospf-state*> indicates the state to which the interface has changed and can be one of the following: • down • attempt • initializing • 2-way • exchange start • exchange • loading • full • unknown

TABLE 446 Syslog messages OSPFv3 (Continued)

Message level	Message	Explanation
Notification	OSPFv3 intf config error, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, error type <*error-type*>, pkt type <*pkt-type*>	Indicates that an OSPF interface configuration error has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the error packet. The <*error-type*> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 446 Syslog messages OSPFv3 (Continued)

Message level	Message	Explanation
Notification	OSPFv3 virtual intf config error, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, error type <*error-type*>, pkt type <*pkt-type*>	Indicates that an OSPF virtual routing interface configuration error has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the error packet. The <*error-type*> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 446 Syslog messages OSPFv3 (Continued)

Message level	Message	Explanation
Notification	OSPFv3 intf authen failure, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, error type <*error-type*>, pkt type <*pkt-type*>	Indicates that an OSPF interface authentication failure has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the authentication failure. The <*error-type*> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 446 Syslog messages OSPFv3 (Continued)

Message level	Message	Explanation
Notification	OSPFv3 virtual intf authen failure, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, error type <*error-type*>, pkt type <*pkt-type*>	Indicates that an OSPF virtual routing interface authentication failure has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the authentication failure. The <*error-type*> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown
Notification	OSPFv3 intf rcvd bad pkt, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, pkt type <*pkt-type*>	Indicates that an OSPF interface received a bad packet. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the authentication failure. The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

TABLE 446 Syslog messages OSPFv3 (Continued)

Message level	Message	Explanation
Notification	OSPFv3 virtual intf rcvd bad pkt, rid <*router-id*>, intf addr <*ip-addr*>, pkt src addr <*src-ip-addr*>, pkt type <*pkt-type*>	Indicates that an OSPF interface received a bad packet. The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*src-ip-addr*> is the IP address of the interface from which the PowerConnect received the authentication failure. The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown
Notification	OSPFv3 intf retransmit, rid <*router-id*>, intf addr <*ip-addr*>, nbr rid <*nbr-router-id*>, pkt type is <*pkt-type*>, LSA type <*lsa-type*>, LSA id <*lsa-id*>, LSA rid <*lsa-router-id*>	An OSPF interface on the PowerConnect has retransmitted a Link State Advertisement (LSA). The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*nbr-router-id*> is the router ID of the neighbor router. The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown The <*lsa-type*> is the type of LSA. The <*lsa-id*> is the LSA ID. The <*lsa-router-id*> is the LSA router ID.
Notification	OSPFv3 virtual intf retransmit, rid <*router-id*>, intf addr <*ip-addr*>, nbr rid <*nbr-router-id*>, pkt type is <*pkt-type*>, LSA type <*lsa-type*>, LSA id <*lsa-id*>, LSA rid <*lsa-router-id*>	An OSPF interface on the PowerConnect has retransmitted a Link State Advertisement (LSA). The <*router-id*> is the router ID of the PowerConnect. The <*ip-addr*> is the IP address of the interface on the PowerConnect. The <*nbr-router-id*> is the router ID of the neighbor router. The <*packet-type*> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown The <*lsa-type*> is the type of LSA. The <*lsa-id*> is the LSA ID. The <*lsa-router-id*> is the LSA router ID.

TABLE 446 Syslog messages OSPFv3 (Continued)

Message level	Message	Explanation
Notification	OSPFv3 originate LSA, rid <*router-id*>, area <*area-id*>, LSA type <*lsa-type*>, LSA id <*lsa-id*>, LSA router id <*lsa-router-id*>	An OSPF interface has originated an LSA. The <*router-id*> is the router ID of the PowerConnect. The <*area-id*> is the OSPF area. The <*lsa-type*> is the type of LSA. The <*lsa-id*> is the LSA ID. The <*lsa-router-id*> is the LSA router ID.
Notification	OSPFv3 max age LSA, rid <*router-id*>, area <*area-id*>, LSA type <*lsa-type*>, LSA id <*lsa-id*>, LSA rid <*lsa-router-id*>	An LSA has reached its maximum age. The <*router-id*> is the router ID of the PowerConnect. The <*area-id*> is the OSPF area. The <*lsa-type*> is the type of LSA. The <*lsa-id*> is the LSA ID. The <*lsa-router-id*> is the LSA router ID.
Notification	OSPFv3 LSDB overflow, rid <*router-id*>, limit <*num*>	A Link State Database Overflow (LSDB) condition has occurred. The <*router-id*> is the router ID of the PowerConnect. The <*num*> is the number of LSAs.
Notification	OSPFv3 LSDB approaching overflow, rid <*router-id*>, limit <*num*>	The software is close to an LSDB condition. The <*router-id*> is the router ID of the PowerConnect. The <*num*> is the number of LSAs.
Notification	OSPFv3 intf rcvd bad pkt Bad Checksum, rid <*ip-addr*>, intf addr <*ip-addr*>, pkt size <*num*>, checksum <*num*>, pkt src addr <*ip-addr*>, pkt type <*type*>	The device received an OSPF packet that had an invalid checksum. The rid <*ip-addr*> is PowerConnect's device ID. The intf addr <*ip-addr*> is the IP address of the interface that received the packet. The pkt size <*num*> is the number of bytes in the packet. The checksum <*num*> is the checksum value for the packet. The pkt src addr <*ip-addr*> is the IP address of the neighbor that sent the packet. The pkt type <*type*> is the OSPF packet type and can be one of the following: • hello • database description • link state request • link state update • link state acknowledgement • unknown (indicates an invalid packet type)
Notification	OSPFv3 intf rcvd bad pkt Bad Packet type, rid <*ip-addr*>, intf addr <*ip-addr*>, pkt size <*num*>, checksum <*num*>, pkt src addr <*ip-addr*>, pkt type <*type*>	The device received an OSPF packet with an invalid type. The parameters are the same as for the Bad Checksum message. The pkt type <*type*> value is "unknown", indicating that the packet type is invalid.

TABLE 446 Syslog messages OSPFv3 (Continued)

Message level	Message	Explanation
Notification	OSPFv3 intf rcvd bad pkt Unable to find associated neighbor, rid <*ip-addr*>, intf addr <*ip-addr*>, pkt size <*num*>, checksum <*num*>, pkt src addr <*ip-addr*>, pkt type <*type*>	The neighbor IP address in the packet is not on the PowerConnect's list of OSPF neighbors. The parameters are the same as for the Bad Checksum message.
Notification	OSPFv3 intf rcvd bad pkt Invalid packet size, rid <*ip-addr*>, intf addr <*ip-addr*>, pkt size <*num*>, checksum <*num*>, pkt src addr <*ip-addr*>, pkt type <*type*>	The device received an OSPF packet with an invalid packet size. The parameters are the same as for the Bad Checksum message.

TABLE 447 Syslog messages IS-IS

Message level	Message	Explanation
Alert	ISIS Memory Limit Exceeded	IS-IS is requesting more memory than is available.
Notification	ISIS ENTERED INTO OVERLOAD STATE	The PowerConnect has set the overload bit to on (1), indicating that the PowerConnect's IS-IS resources are overloaded.
Notification	ISIS Entered Overload State Due to <*overload-reason*>	The PowerConnect has set the overload bit to on (1), indicating that the PowerConnect's IS-IS resources are Overloaded. Reasons for the overload as expressed in the <*overload-reason*> variable are: • Configuration • Startup Configuration • LSP Buffer Allocation Failure • LSP Header Allocation Failure • Maximum Number of LSPs Exceeded • LSP Fragmentation Count Exceeded • LSP Sequence Number Wrap Around • LSP Option Allocation Failure • Path Entry Allocation Failure • Route Entry Allocation Failure Definitions of the <*overload-reason*> values are described in Table 448.
Notification	ISIS Exited Overload State	The PowerConnect has set the overload bit to off (0), indicating that the PowerConnect's IS-IS resources are no longer overloaded.
Notification	ISIS L1 ADJACENCY DOWN <*system-id*> on circuit <*circuit-id*>	The PowerConnect's adjacency with this Level-1 IS has gone down. The <*system-id*> is the system ID of the IS. The <*circuit-id*> is the ID of the circuit over which the adjacency was established.

TABLE 447 Syslog messages IS-IS (Continued)

Message level	Message	Explanation
Notification	ISIS L1 ADJACENCY UP <*system-id*> on circuit <*circuit-id*>	The PowerConnect's adjacency with this Level-1 IS has come up. The <*system-id*> is the system ID of the IS. The <*circuit-id*> is the ID of the circuit over which the adjacency was established.
Notification	ISIS L2 ADJACENCY DOWN <*system-id*> on circuit <*circuit-id*>	The PowerConnect's adjacency with this Level-2 IS has gone down. The <*system-id*> is the system ID of the IS. The <*circuit-id*> is the ID of the circuit over which the adjacency was established.
Notification	ISIS L2 ADJACENCY UP <*system-id*> on circuit <*circuit-id*>	The PowerConnect's adjacency with this Level-2 IS has come up. The <*system-id*> is the system ID of the IS. The <*circuit-id*> is the ID of the circuit over which the adjacency was established.
Notification	ISIS <*LSP-type*> LSP <*LSP-ID*> Seq <*sequence-number*> Len <*length*> LifeTime <*lifetime*> on <*interface-name*> dropped due to <*LSP-drop-reason*>	The PowerConnect has dropped the received LSP. The <*LSP-Type*> can be one of the following: • L1 • L2 The <*LSP-ID*> variable is in the 8 byte LSP ID value. The <*sequence-number*> is a 4 byte value that is associated with each LSP ID. The <*length*> is the length of the LSP PDU. The <*lifetime*> is the life period of the LSP. The <*interface-name*> is the name of the interface and is displayed in the following form "Ethernet 1/1". The <*LSP-drop-reason*> variable describes the following reasons that the LSP was dropped: • Adjacency not found • Adjacency Level Mismatch • IS Level Mismatch • Length Too Short • Length Too Large • Authentication Failure • Max Area Check Failure • Zero Checksum • Checksum Mismatch • Invalid Length Definitions of the <*LSP-drop-reason*> values are described in Table 448.

TABLE 447 Syslog messages IS-IS (Continued)

Message level	Message	Explanation
Notification	ISIS <*NbrType*> Neighbor <*Hostname/systemID*> DOWN on <*interface-name*> due to <*neighbor-down-reason*>	The PowerConnect's Neighbor has gone down. The <*NbrType*> can be one of the following: • L1 • L2 • PTPT The <*interface-name*> is the name of the interface and is displayed in the following form "Ethernet 1/1". The <*neighbor-down-reason*> variable can be any one of the following reasons that the Neighbor is Down: • BFD Trigger • Maximum Adjacencies • User Trigger • Hold Timer Expiry • Adjacency ID Mismatch • Adjacency Type Mismatch • Interface Down • Interface State Change Definitions of the <*neighbor-down-reason*> values are described in Table 448.
Notification	ISIS <*NbrType*> neighbor <*Hostname/systemID*> UP on <*interface-name*>	The PowerConnect's Neighbor has come up. The <*NbrType*> can be one of the following: • L1 • L2 • PTPT The <*interface-name*> is the name of the interface and is displayed in the following form "Ethernet 1/1".
Notification	ISIS PTP ADJACENCY DOWN <*mac*> on interface <*portnum*>	
Notification	ISIS PTP ADJACENCY UP <*mac*> on interface <*portnum*>	

TABLE 448 Definition of IS-IS variables

Variable	Value	Definition
<*neighbor-down-reason*>	BFD Trigger	BFD identified link failures and triggered IS-IS to clean the neighbors on that link.
	Maximum Adjacencies	IS-IS has reached the maximum number of adjacencies. Therefor, it has deleted the adjacency with the lowest SNPA address to accommodate the new adjacency.
	User Trigger	The user triggered to delete the adjacency using the clear **isis neighbor** <*systemID*> command or the **clear isis all** command.
	Hold Timer Expiry	The adjacency was deleted because there were no "hellos" received within the hold time period.

TABLE 448 Definition of IS-IS variables (Continued)

Variable	Value	Definition
	Adjacency ID Mismatch	The adjacency was deleted because the new “hello” received from this adjacency has a different System ID.
	Adjacency Type Mismatch	The adjacency was deleted because the new “hello” received from this adjacency has a different adjacency Type.
	Interface Down	The adjacency was deleted because the interface went down.
	Interface State Change	The adjacency was deleted because the interface state has changed due to user configuration.
<overload-reason>	Configuration	The Overload condition was entered because of a user configuration.
	Startup Configuration	The Overload condition was entered because of the startup configuration.
	LSP Buffer Allocation Failure	The Overload condition was entered because of an LSP buffer allocation error.
	LSP Header Allocation Failure	The Overload condition was entered because of an LSP header allocation error.
	Maximum Number of LSPs Exceeded	The Overload condition was entered because the LSP count reached the maximum value.
	LSP Fragmentation Count Exceeded	The Overload condition was entered because of IS-IS trying to generate the 256th LSP fragment.
	LSP Sequence Number Wrap Around	The Overload condition was entered because the LSP numbers reached the maximum value.
	LSP Option Allocation Failure	Self LSP building failed due to an internal buffer allocation failure.
	Path Entry Allocation Failure	The SPF computation failed due to a Path Entry allocation failure.
	Route Entry Allocation Failure	The SPF computation failed due to a Route Entry allocation failure.
<LSP-drop-reason>	Adjacency not found	The LSP was dropped because the is no adjacency found on the interface.
	Adjacency Level Mismatch	The LSP was dropped because the adjacency is at a different level from the LSP level.
	IS Level Mismatch	The LSP was dropped because IS-IS is configured at a different level than the LSP level.
	Length Too Short	The LSP length is shorter than the LSP header length.
	Length Too Large	The LSP length is larger than the Maximum LSP buffer length.
	Authentication Failure	The LSP was dropped because of an authentication failure.
	Max Area Check Failure	The LSP has a Max Area Count different than the configured Max Area Count of the device.
	Zero Checksum	The LSP has a zero checksum.

TABLE 448 Definition of IS-IS variables (Continued)

Variable	Value	Definition
	Checksum Mismatch	The LSP checksum is different than the computed checksum.
	Invalid Length	The LSP length is different than the sum of the option lengths in the LSP.

TABLE 449 Syslog messages BGP

Message level	Message	Explanation
Debug	BGP4 Not enough memory available to run BGP4	The device could not start the BGP4 routing protocol because there is not enough memory available.
Error	BGP No of prefixes received from BGP peer <*ip-addr*> exceeds maximum prefix-limit...shutdown	The PowerConnect has received more than the specified maximum number of prefixes from the neighbor, and the PowerConnect is therefore shutting down its BGP4 session with the neighbor.
Error	BGP received invalid AS4_PATH attribute length (3) - entire AS4_PATH ignored	Possible attribute length can be only even number and cannot be odd. If an attribute with odd length is received, this error is displayed.
Error	BGP received invalid AS4_PATH attribute flag (0x40) - entire AS4_PATH ignored	If the flag that describes the attribute has unacceptable values then this error is displayed.
Error	BGP received invalid Confed info in AS4_PATH (@byte 43) - entire AS4_PATH ignored	Confedration segments(AS_CONFED_SEQ/SET) must precede the (AS_SEQ/SET), if not, this error is displayed.
Error	BGP received incorrect Seq type/len in AS4_PATH (@byte 41) - entire AS4_PATH ignored	Valid segment types are (AS_SEQ/SET, AS_CONFED_SEQ/SET), any other values results in an error being displayed.
Error	BGP received multiple AS4_PATH attributes - used first AS4_PATH attribute only	When AS4_PATH is received more than one time in the update message, this error is displayed.
Warning	BGP No of prefixes received from BGP peer <*ip-addr*> exceeds warning limit <*num*>	The PowerConnect has received more than the allowed percentage of prefixes from the neighbor. The <*ip-addr*> is the IP address of the neighbor. The <*num*> is the number of prefixes that matches the percentage you specified. For example, if you specified a threshold of 100 prefixes and 75 percent as the warning threshold, this message is generated if the PowerConnect receives a 76th prefix from the neighbor.
Notification	BGP Peer <*ip-addr*> UP (ESTABLISHED)	Indicates that a BGP4 neighbor has come up. The <*ip-addr*> is the IP address of the neighbor's BGP4 interface with the PowerConnect.

TABLE 449 Syslog messages BGP (Continued)

Message level	Message	Explanation
Notification	BGP Peer <*ip-addr*> DOWN (IDLE)	Indicates that a BGP4 neighbor has gone down. The <*ip-addr*> is the IP address of the neighbor's BGP4 interface with the PowerConnect.
Notification	BGP Peer <*ip*> DOWN (<*reason*><*recv notif*>)	
Notification	Configuration (Wait for BGP)	IS-IS is waiting for BGP convergence to complete. (See "Setting the overload bit" on page 941.)

TABLE 450 Syslog messages NTP

Message level	Message	Explanation
Warning	NTP server <*ip-addr*> failed to respond	Indicates that a Simple Network Time Protocol (SNTP) server did not respond to the device's query for the current time. The <*ip-addr*> indicates the IP address of the SNTP server.

TABLE 451 Syslog messages TCP

Message level	Message	Explanation
Notification	TCP Local TCP exceeds <*burst-max*> burst packets, stopping for <*lockup*> seconds!!	The number of TCP SYN packets exceeds the <*burst-max*> threshold set by the **ip tcp burst** command. The PowerConnect may be the victim of a TCP SYN DoS attack. All TCP SYN packets will be dropped for the number of seconds specified by the <*lockup*> value. When the lockup period expires, the packet counter is reset and measurement is restarted.
Notification	TCP Transit TCP in interface <*portnum*> exceeds <*num*> burst packets, stopping for <*num*> seconds!!	Threshold parameters for TCP transit (through) traffic have been configured on an interface, and the maximum burst size for TCP packets on the interface has been exceeded. The <*portnum*> is the port number. The first <*num*> is the maximum burst size (maximum number of packets allowed). The second <*num*> is the number of seconds during which additional TCP packets will be blocked on the interface. **NOTE:** This message can occur in response to an attempted TCP SYN attack.

TABLE 452 Syslog messages DOT1X

Message level	Message	Explanation
Warning	DOT1X security violation at port <*portnum*>, malicious mac address detected <*mac-address*>	A security violation was encountered at the specified port number.
Warning	DOT1X Port <*portnum*>, AuthControlledPortStatus change restricted	
Notification	DOT1X Port <*portnum*> port default vlan-id changes to <*vlan-id*>	
Notification	DOT1X Port <*portnum*> currently used vlan-id changes to <*vlan-id*> due to move to restricted vlan	
Notification	DOT1X issues software port up indication of Port <*portnum*> to other software applications	The device has indicated that the specified port has been authenticated, but the actual port may not be active.
Notification	DOT1X issues software port down indication of Port <*portnum*> to other software applications	The device has indicated that the specified is no longer authorized, but the actual port may still be active.
Informational	DOT1X Port <*portnum*>, AuthControlledPortStatus change authorized	The status of the interface's controlled port has changed from unauthorized to authorized.
Informational	DOT1X Port <*portnum*>, AuthControlledPortStatus change unauthorized	The status of the interface's controlled port has changed from authorized to unauthorized.
Informational	DOT1X Port <*portnum*> currently used vlan-id changes to <*vlan-id*> due to dot1x-RADIUS vlan assignment	A user has completed 802.1X authentication. The profile received from the RADIUS server specifies a VLAN ID for the user. The port to which the user is connected has been moved to the VLAN indicated by <*vlan-id*>.
Informational	DOT1X Port <*portnum*> currently used vlan-id is set back to port default vlan-id <*vlan-id*>	The user connected to <*portnum*> has disconnected, causing the port to be moved back into its default VLAN, <*vlan-id*>.
Informational	DOT1X Port <*portnum*> is unauthorized because system resource is not enough or the invalid information to set the dynamic assigned IP ACLs or MAC address filters	802.1X authentication could not take place on the port. This happened because strict security mode was enabled and one of the following occurred: • Insufficient system resources were available on the device to apply an IP ACL or MAC address filter to the port • Invalid information was received from the RADIUS server (for example, the Filter-ID attribute did not refer to an existing IP ACL or MAC address filter)
Debug	DOT1X Not enough memory	There is not enough system memory for 802.1X authentication to take place. Contact Dell Technical Support.

TABLE 453 Syslog messages SNMP

Message level	Message	Explanation
Informational	SNMP Auth. failure, intruder IP <*ip-addr*>	A user has tried to open a management session with the device using an invalid SNMP community string. The <*ip-addr*> is the IP address of the host that sent the invalid community string.
Informational	SNMP read-only community | read-write community | contact | location | user | group | view | engineId | trap [host] [<*value -str*>] deleted | added | modified from console | telnet | ssh | web | snmp session	A user made SNMP configuration changes through the Web, SNMP, console, SSH, or Telnet session. [<*value-str*>] does not appear in the message if SNMP community or engineId is specified.

TABLE 454 Syslog messages MPLS

Message level	Message	Explanation
Informational	Deleting VLL <*name*> (ID <*number*>)at <*string*> port <*slot/port*> with peer IPv4 address <*ip-address*>	Sent when PW traps are generated if the PW has been deleted, i.e. when the pwRowStatus in the MIB has been set to destroy(6), the PW has been deleted by a non-MIB application, or due to auto discovery process.
Informational	MPLS Deleting VLL <*vll-name*> (ID <*vll-id*>)	Sent when the specified VLL is being deleted.
Informational	MPLS Deleting VLL <*vll-name*> (ID <*vll-id*>) at {tagged | untagged} port <*slot*>/<*port*>	Sent when the specified VLL with the at the specified tagged or untagged port is being deleted.
Informational	MPLS Deleting VLL <*vll-name*> (ID <*vll-id*>) with peer IPv4 address <*ip*>	Sent when the specified VLL with the specified IPv4 peer is being deleted.
Informational	VLL is down for table index <*number*>	Sent when PW traps are generated if the VLL is down for one index
Informational	VLL is up for table index <*number*>	Sent when PW traps are generated if VLL is up for one index
Informational	VLLs are down for table indexes <*number*> through <*number*>	Sent when PW traps are generated if the VLLs represented by sequential entries in the database are down
Informational	VLLs are up for table indexes <*number*> through <*number*>	Sent when PW traps are generate dif VLLs represented by sequential entries in the database are up
Informational	VRF Port <*slot-port*> added to VRF <*name*> with updated port count <*number*>	Sent when an MPLS L3 VPN trap is generated if the state of an interface within the VRF changed from down to up.
Informational	VRF Port <*slot-port*> deleted from VRF <*name*> with updated port count <*number*>	Sent when an MPLS L3 VPN trap is generated if the state of an interface within the VRF changed from down to up.
Notification	MPLS Deleting VLL <*name*> (ID <*vc-id*>) at {tagged I untagged} port <*portnum*> with peer IPv4 address <*ip*>	

TABLE 454 Syslog messages MPLS (Continued)

Message level	Message	Explanation
Notification	MPLS LSP <*lspname*> switches to new active path <*pathame*>	
Notification	MPLS LSP <*lspname*> using path <*pathname*> is down	
Notification	MPLS LSP <*lspname*> using path <*pathname*> is up	
Notification	MPLS VLL is down for table index <*n*>	
Notification	MPLS VLL is up for table index <*n*>	
Notification	MPLS VLLs are down for table indexes <*n*> through <*m*>	
Notification	MPLS VLLs are up for table indexes <*n*> through <*m*>	
Notification	MPLS VPLS [ID <*id*>] peer <*ip*> is down	Sent when a single VPLS peer is transitioning to a down state
Notification	MPLS VPLS [ID <*id*>] peer <*ip*> is up	Sent when a single VPLS peer is transitioning to an up state
Notification	MPLS VPLS <*name*> (ID <*id*>) endpoint <*ip-address*> is down	Sent when a single VPLS endpoint is transitioning to a down state.
Notification	MPLS VPLS <*name*> (ID <*id*>) endpoint <*ip-address*> is up	Sent when a single VPLS endpoint is transitioning to an up state.
Notification	MPLS VPLS for instance indices <*list*> <*n*> through <*m*> are up	Sent when multiple VPLS instances are transitioning to an up state.
Notification	MPLS VPLS for instance indices <*list*> <*n*> through <*m*> are down	Sent when multiple VPLS instances are transitioning to a down state.
Notification	MPLS VPLS peer <*ip*> associated with VC ID <*n*> is up	Sent when a single VPLS peer is transitioning to an up state.
Notification	MPLS VPLS peer <*ip*> associated with VC ID <*n*> is down	Sent when a single VPLS peer is transitioning to a down state.
Notification	MPLS VPLS peer <*ip*> associated with instances <*n*>-<*m*> <*list*> is down	Sent when multiple VPLS instances associated with a peer are transitioning to a down state.
Notification	MPLS VPLS peer <*ip*> associated with instances <*list*> <*n*>-<*m*> <*list*> is up	Sent when multiple VPLS instances associated with a peer are transitioning to an up state.
Notification	MPLS VPL endpoint <*slot*>/<*port*> associated with instance indices <*list*> is dowm	Sent when multiple VPLS instances associated with an endpoint is transitioning to a down state.
Notification	MPLS VPL endpoint <*slot*>/<*port*> associated with instance indices <*list*> is up	Sent when multiple VPLS instances associated with an endpoint is transitioning to an up state.
Notification	MPLS VLL-Local <*name*> is down	Sent when a single VLL-Local instance is transitioning to a down state.
Notification	MPLS VLL-Local <*name*> is up	Sent when a single VLL-Local instance is transitioning to an up state.

TABLE 454 Syslog messages MPLS (Continued)

Message level	Message	Explanation
Notification	MPLS VLL-Local for instance indices <*list*> <*n*> through <*m*> are up	Sent when multiple VLL-Local instances are transitioning to an up state.
Notification	MPLS VLL-Local for instance indices <*list*> <*n*> through <*m*> are down	Sent when multiple VLL-Local instances are transitioning to a down state.
Notification	MPLS VLL <*name*> (ID <*id*> is down	Sent when a single VLL peer is transitioning to a down state.
Notification	MPLS VLL <*name*> (ID <*id*> is up	Sent when a single VLL peer is transitioning to an up state.
Notification	MPLS VLL for instance indices <*list*> <*n*> through <*m*> are up	Sent when multiple VLL instances are transitioning to an up state.
Notification	MPLS VLL for instance indices <*list*> <*n*> through <*m*> are down	Sent when multiple VLL instances are transitioning to a down state.

TABLE 455 Syslog messages VRF

Message level	Message	Explanation
Notification	VRF Port <*portnum*> added to VRF <*name*> with updated port count <*n*>	
Notification	VRF Port <*portnum*> deleted from VRF <*name*> with updated port count <*n*>	
Informational	VRF <vrf_name> has been configured as management VRF.	Indicates that the specified VRF has been configured as a management VRF.
Informational	VRF <vrf_name> has been un-configured as management VRF.	Indicates that the specified VRF has been removed as a management VRF.

TABLE 456 Syslog messages

Message level	Message	Explanation
Notification	Authentication Enabled on <*portnum*>	The multi-device port authentication feature was enabled on the on the specified <*portnum*>.
Notification	Authentication Disabled on <*portnum*>	The multi-device port authentication feature was disabled on the on the specified <*portnum*>.

TABLE 457 Syslog messages BFD

Message level	Message	Explanation
Notification	BFD Session UP for NBR <*neighbor-ID*> on <*port*>	The BFD session is UP with the neighbor specified by the <*neighbor-ID*> on the port specified by the <*port*> variable.

TABLE 457 Syslog messages BFD (Continued)

Message level	Message	Explanation
Notification	BFD Session DOWN for NBR <*neighbor-ID*> on <*port*> Reason Neighbor Signaled Session Down	The BFD session with the neighbor specified by the <*neighbor-ID*> on the port specified by the <*port*> variable is Down because the BFD neighbor has signaled the session to be down.
Notification	BFD Session DOWN for NBR <*neighbor-ID*> on <*port*> Reason Administratively Down	The BFD session with the neighbor specified by the <*neighbor-ID*> on the port specified by the <*port*> variable is Down for Administrative reasons.

TABLE 458 Syslog messages Optics

Message level	Message	Explanation
Notification	Transceiver type checking has been disabled!	The transceiver type checking feature has been disabled. The device will continue to report incompatible transceivers through syslog messages and but will not shutdown a port that contains one.
Notification	Session DOWN for LSP <*lsp-name*> Reason Adminstratively Down	The BFD session for the LSP specified by the <*lsp-name*> is Down for Administrative reasons.
Notification	Session Up for LSP <*lsp-name*>	The BFD session for the LSP specified by the <*lsp-name*> is Up.
Notification	Session DOWN for RSVP session <*session-id*> Reason Adminstratively Down	The BFD session for the RSVP session specified by the <*session-id*> is Down for Administrative reasons. The form of the <*session-id*> displayed is IPv4 tunnel endpoint or tunnel ID or extended tunnel ID. For example 22.22.22.2/3/11/11/11/1
Notification	Session UP for RSVP session <*session-id*>	The BFD session for the RSVP session specified by the <*session-id*> is Up. The form of the <*session-id*> displayed is IPv4 tunnel endpoint or tunnel ID or extended tunnel ID. For example 22.22.22.2/3/11/11/11/1
Notification	Transceiver type checking has been enabled!	The transceiver type checking feature has been re-enabled. The feature is enabled by default and does not send the message under normal circumstances. However, if it is disabled and then re-enabled the device will send this message..
Warning	Optic is not Dell qualified (<*port*>) Type <*type-description*> Vendor <*vendor-name*> , Version <*version-num*> Part# <*part-no*>, Serial# <*serial-no*>	The optic module installed in the Interface module at the port specified by the <*port*> variable is not Dell qualified although the port is still operational. The Type, Vendor, Version, Part#, and Serial # of the optic module is provided.

TABLE 458 Syslog messages Optics (Continued)

Message level	Message	Explanation
Alert	Optic is not Dell qualified, optical monitoring is not supported (<*port*>) Type <*type-description*> Vendor <*vendor-name*> , Version <*version-num*> Part# <*part-no*>, Serial# <*serial-no*>	The optic module installed in the Interface module at the port specified by the <*port*> variable is not Dell qualified and will not be able to be monitored using the Optical Monitoring function. The Type, Vendor, Version, Part#, and Serial # of the optic module is provided.
Alert	Optic is not capable of optical monitoring (<*port*>) Type <*type-description*> Vendor <*vendor-name*> , Version <*version-num*> Part# <*part-no*>, Serial# <*serial-no*>	The optic module installed in the Interface module at the port specified by the <*port*> variable is not able to be monitored using the Optical Monitoring function. The Type, Vendor, Version, Part#, and Serial # of the optic module is provided.
Alert	Incompatible optical trans-receiver detected on port <*n*>	Indicates that in incompatible XFP or SFP has been installed in the port specified. A port with an incompatible optical module installed are shut down.

TABLE 459 Syslog messages LDP

Message level	Message	Explanation
Notification	MPLS LDP path vector limit mismatch for session <*lsrId*><*labelSpaceId*> (value <*local vector limit*>) with peer <*lsrId*><*labelSpaceId*> (value <*peer vector limit*>)	This notification is generated when the value of the LDP path vector limit value from the peer does not match that of the entity.
Notification	MPLS LDP entity session <*lsrId*><*labelSpaceId*> with peer <*lsrId*><*labelSpaceId*> is up	This notification is sent when the value of 'mplsLdpSessionState' enters the 'operational(5)' state.
Notification	MPL LDP entity session <*lsrId*><*labelSpaceId*> with peer <*lsrId*><*labelSpaceId*> is down	This notification is sent when the value of 'mplsLdpSessionState' leaves the 'operational(5)' state.

A Syslog messages

Appendix

Global and Address Family Configuration Levels

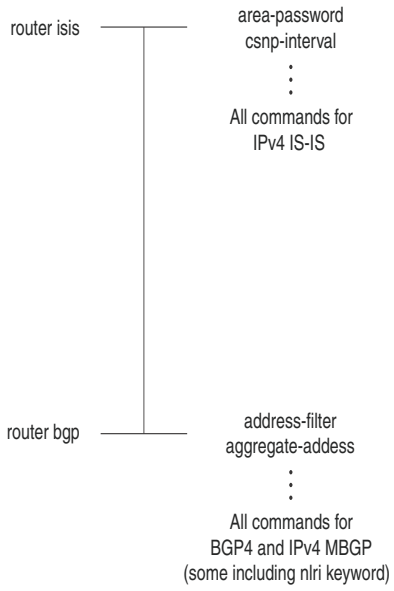
B

The following Global and Address Family Configuration Level features are supported by the NetIron MLX Series devices.

- Accessing the Address Family Configuration Level Under BGP
- Backward Compatibility for Existing BGP4 and IPv4 IS-IS
- Global BGP4 Commands and BGP4 Unicast Route Commands

This appendix describes the restructured CLI for BGP and IS-IS on devices that support IPv6. In earlier versions of software, the CLI for BGP4 and IPv4 IS-IS is structured as shown in Figure 240.

FIGURE 240 Earlier structure of BGP4 and IPv4 IS-IS CLI



To configure BGP4 and IPv4 MBGP, enter the **router bgp** command, which takes you to the BGP router configuration level. At this level, you can access commands to configure all aspects of BGP4 and IPv4 MBGP, including commands that configure the protocol, and commands that configure unicast routes and multicast routes. (To configure aspects of multicast routes, specify the nlri keyword with a command.)

To configure IPv4 IS-IS, enter the **router isis** command, which takes you to the IS-IS device configuration level. At this level, you can access commands that allow you to configure all aspects of IPv4 IS-IS, including commands that configure the protocol, and commands that configure unicast routes.

In both cases, the device determines, for example, whether commands entered at the BGP device configuration level apply to BGP4, to BGP4 unicast routes, or to IPv4 MBGP routes.

The introduction of IPv6 required the restructuring of existing BGP4 and IPv4 IS-IS CLI for the following reasons:

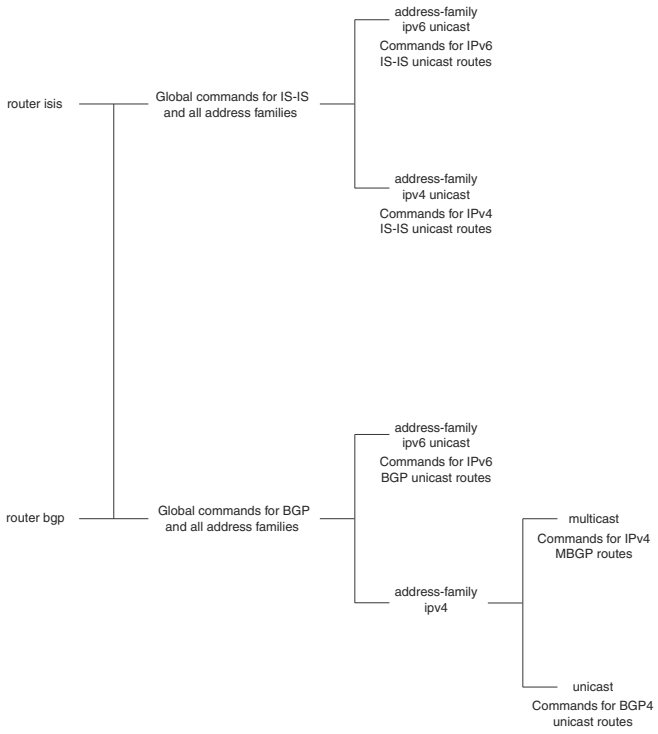
- To accommodate the IPv6-related CLI.
- To simplify the configuration of BGP4 unicast and IPv4 MBGP routes.

The CLI includes two layers of CLI for BGP and IS-IS (refer to Figure 241):

- A global layer to configure BGP and IS-IS protocols.
- Address families that separate the configuration of:
  - IPv6 and IPv4
  - Routing protocol

  Sub-address families separate the configuration of:
  - Unicast routes
  - Multicast routes

FIGURE 241 IPv4, BGP4+ and IS-IS CLI structures



Accessing the address family configuration level

For example, to access the BGP4 multicast address family configuration level, enter the following command while at the global BGP configuration level.

```
NetIron(config-bgp)# address-family ipv4 multicast
NetIron(config-bgp-ipv4m)#
```

Syntax: address-family ipv4 unicast | ipv4 multicast | ipv6 unicast

The (config-bgp-ipv4m)# prompt indicates that you are at the IPv4 multicast address family configuration level. At this level, you can access commands that allow you to configure BGP4 multicast routes. The commands that you enter at this level apply to BGP4 multicast routes only. You can generate a configuration for BGP4 multicast routes that is separate and distinct from configurations for BGP4 unicast routes and BGP4+ unicast routes.

NOTE
Each address family configuration level gives you access to commands that apply to that address family only. To enable a feature in a particular address family, you must specify any associated commands for that feature in that address family.

To exit the BGP4 multicast address family configuration level, enter this command.

```
NetIron(config-bgp-ipv4m)# exit-address-family
NetIron(config-bgp)#
```

Syntax: exit-address-family

When you enter the **exit-address-family** command at an address family configuration level you return to the global IS-IS configuration level, or the BGP4 unicast address family configuration level, (the default BGP4 level). For backward compatibility, you can currently access commands related to BGP4 unicast routes at both global BGP4 configuration and BGP4 unicast address family configuration levels. Both levels are indicated by the (config-bgp)# prompt.

Backward compatibility for existing BGP4 and IPv4 IS-IS configurations

When a device is upgraded to the current software version, the software automatically converts the existing BGP4 unicast and all IPv4 IS-IS configurations into the new address families. The software also automatically converts some of the IPv4 MBGP configuration into the new address family. Software conversion actions include:

- Leaves the global BGP4 and IPv4 IS-IS configurations as is.
- Converts the configuration for BGP4 unicast neighbors and routes into the BGP4 unicast address family.
- Converts the configuration for IPv4 IS-IS unicast routes into the IPv4 IS-IS unicast address family.
- Converts the configuration for IPv4 MBGP neighbors into IPv4 MBGP address family.

NOTE
The software does not convert all aspects of the IPv4 MBGP configuration. You must reconfigure the network routes, aggregate routes, redistribution of routes into IPv4 MBGP, and route map filters. Use the **show run** and **show ip bgp config** commands to check your IPv4 MBGP configuration.

Previously, IPv4 MBGP routes were configured using commands that included the **nlri** keyword The current software version does not support the **nlri** keyword with IPv4 and IPv6 MBGP commands. You must now use the address families to configure all versions of BGP, IS-IS, and MBGP.

Global BGP4 commands and BGP4 unicast route commands

A global BGP command configures the BGP routing protocol and applies to all IPv4 and IPv6 address families. You can access the global commands while at the global BGP configuration level.

A BGP4 unicast route command configures a BGP4 unicast route. For backward compatibility, you can access BGP4 unicast route commands at both global BGP4 configuration and BGP4 unicast address family configuration levels. To help you distinguish the global BGP4 commands from the BGP4 unicast route commands at the global BGP4 configuration level, this section lists global BGP commands:

- **address-filter**
- **always-compare-med**
- **as-path-filter**
- **as-path-ignore**
- **bgp-redistribute-internal**
- **cluster-id**
- **community-filter**
- **compare-routerid**
- **confederation identifier**
- **confederation peers**
- **default-local-preference**
- **distance**
- **enforce-first-as**
- **fast-external-fallover**
- **ignore-invalid-confed-as-path**
- **local-as**
- **med-missing-as-worst**
- **timers**

The following global BGP commands are used to configure peer groups and neighbors:

- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **description**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **distribute-list acl_name in**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **distribute-list acl_name out**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **distribute-list in**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **distribute-list out**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **ebgp-multihop**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **filter-list in**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **filter-list out**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **next-hop-self**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **password**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **peer-group**

- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **remote-as**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **remove-private-as**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **shut_down**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **soft-reconfiguration inbound**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **timers**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **update-source**

The following address family commands modify the behavior of BGP for a specific address family:

- **aggregate-address**
- **client-to-client-reflection**
- **dampening**
- **default-information-originate**
- **default-metric**
- **maximum-paths**
- **multipath**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **filter-list in** (applies to the IPv4 unicast address family only)
- **network**
- **next-hop-enable-default**
- **next-hop-recursion** (applies to the IPv4 unicast address family only)
- **readvertise** (applies to the IPv4 unicast address family only)
- **redistribute**
- **table-map**
- **update-time**

The following commands configure policies for neighbors or peer groups for a specific address family:

- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **activate**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **capability orf prefixlist**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **default-originate**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **filter-list as-path-access-list in**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **filter-list as-path-access-lis out**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **maximum-prefix**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **prefix-list prefix_list_name in**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **prefix-list prefix_list_name out**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **route-map in**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **route-map out**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **route-reflector-client**

- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **send-community**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **unsuppress-map**
- **neighbor** <*ipv4-address*> | <*ipv6-address*> | <*peer-group-name*> **weight**

Currently, you can create a neighbor with an IPv4 or IPv6 address at the global BGP configuration or IPv4 unicast address family configuration level. For example, if you create a neighbor with an IPv4 address at this level, by default, the neighbor is enabled to exchange IPv4 unicast prefixes. However, this neighbor cannot exchange IPv4 multicast prefixes until you explicitly enable it to do so by entering the **neighbor** <*ipv4-address*> | <*peer-group-name*> **activate** command at the IPv4 multicast address family configuration level. Likewise, if you create a neighbor with an IPv6 address at this level, the neighbor will not exchange IPv6 unicast prefixes until you explicitly enable it to do so by entering the **neighbor** <*ipv6-address*> | <*peer-group-name*> **activate** command at the IPv6 unicast address family configuration level.

If you create a neighbor at the IPv4 multicast address family configuration or IPv6 unicast address family configuration levels, by default, the neighbor is enabled to exchange IPv4 multicast prefixes or IPv6 unicast prefixes, respectively. You do not need to explicitly enable the neighbor at either level.

Appendix

Commands That Require a Reload

C

The following commands that require a reload features are supported by the NetIron MLX Series devices.

- cam-mode ip
- cam-mode ipvpn
- default-max-frame-size
- multicast-flooding
- port-priority
- system-max
- virtual-interface-mac
- vll-mtu-enforcement

Most CLI commands take effect as soon as you enter them. However, a small number of commands require a software reload to take effect. Table 460 lists these commands.

To place a configuration change made by one of these commands into effect, you must save the change to the startup-config file, then reload the software. If you reload the software without saving the change to the startup-config file, the device does not make the change.

To reload the software, you must perform a warm start. To perform a warm start, do one of the following:

- Enter the **reload** command at the Privileged EXEC level of the CLI.
- Enter the **boot system** command at the Privileged EXEC level of the CLI.

TABLE 460 Commands that require a software reload

Command	See ...
cam-mode ip	"Configuring FDR globally" on page 2239
cam-mode ipvpn	"Configuring FDR globally" on page 2239
default-max-frame-size	"Setting the maximum frame size globally" on page 681
multicast-flooding	"Hardware flooding for layer 2 multicast and broadcast packets" on page 261
system-max	"Displaying and modifying default settings for system parameters" on page 100
virtual-interface-mac	"Assigning a MAC address to a virtual interface" on page 677
vll-mtu-enforcement	"Enabling VLL MTU enforcement (optional)" on page 1466

Appendix

Software Specifications

D

This appendix lists the following information for the PowerConnect Series devices:

- IPv6 Phase II
- IEEE compliance
- RFC support
- Internet draft support

IPv6 Phase II

	The PowerConnect Series devices are compliant to IPv6 Phase II specifications.

IEEE COMPLIANCE

- 802.1ag — Connectivity Fault Management
- 802.1D — MAC Bridges
- 802.1Q — Virtual Bridged LANs
- 802.1s — Multiple Spanning Trees
- 802.1w — Rapid STP
- 802.1X — User authentication
- 802.3 — Ethernet Like MIB
- 802.3ad — Link Aggregation
- 802.3ae —10-Gigabit Ethernet
- 802.3x — Flow Control
- Ethernet Interface MIB
- Repeater MIB
- SNMP MIB II
- SNMP v1, v2c and V3

RFC COMPLIANCE

RFC compliance - BGPv4

- 1745 — OSPF Interactions
- 1772 — Application of BGP in the Internet
- 1997 — Communities & Attributes
- 2385 — BGP Session Protection viaTCP MD5
- 2439 — Route Flap Dampening
- 4456 — Route Reflection
- 2918 — Route Refresh Capability
- 3065 — BGP4 Confederations
- 3392 — Capabilities Advertisement with BGP-4
- 3682 — Generalized TTL Security Mechanism, for eBGP Session Protection
- 4271 — BGPv4
- 4273 — BGP MIB
- 4486 — Subcodes for BGP Cease Notification Message
- 4893 — BGP Support for Four-octet AS Number Space

RFC compliance - OSPF

- 1745 — OSPF Interactions
- 1765 — OSPF Database Overflow
- 1850 — OSPF Traps
- 2154 — OSPF w/Digital Signatures (Password, MD-5)
- 2328 — OSPF v2
- 2370 — OSPF Opaque LSA Option (partially supported)
- 3101 — OSPF NSSA
- 3137 — OSPF Stub Router Advertisement
- 3623 — Graceful OSPF Restart
- 3630 — TE Extensions to OSPF v2
- 4303 — IP Encapsulating Security Payload (ESP)
- 4552 — Authentication/Confidentiality for OSPFv3
- 4835 — Cryptographic Message Syntax (CMS) Multiple Signer Clarification

RFC compliance - IS-IS

- 1142 — OSI IS-IS Intra-domain Routing Protocol
- 1195 — Routing in TCP or IP and Dual Environments
- 2763 — Dynamic Host Name Exchange

- 2966 — Domain-wide Prefix Distribution
- 2973 — IS-IS Mesh Groups
- 3567 — IS-IS Cryptographic Authentication (MD-5) (partially supported)
- 3373 — Three-Way Handshake for IS-IS Point-to-Point Adjacencies

RFC compliance - RIP

- 1058 — RIP v1
- 1723 — RIP v2
- 1812 — RIP Requirements

RFC compliance - IP multicast

- 1075 — DVMRP v2
- 1112 — Host Extensions
- 2362 — PIM-SM
- 2858 — BGP-MP
- 3376 — IGMP v3
- 3446 — Anycast RP
- 4610— Anycast-RP Using Protocol Independent Multicast (PIM)
- 3569 — Overview of SSM
- 3618 — MSDP
- 3973 — PIM-DM
- 4611— MSDP Deployment Scenarios
- PIM-DM v1

RFC compliance - general protocols

- 1027 — Proxy ARP
- 1042—Standard for the Transmission of IP Datagrams over IEEE 802 Networks
- 1122 — Host Requirements
- 1166 — Internet Numbers
- 1256 — IRDP
- 1332— PPP IPCP
- 1340 — Assigned Numbers
- 1354 — IP Forwarding Table MIB
- 1377— PPP OSI NLCP
- 1519 — CIDR
- 1542 — BootP Extensions
- 1591 — DNS (client)
- 1661 — PPP

- 1662 — PPP in HDLC Framing
- 1812 — Requirements for IPv4 Routers
- 1858 — Security Considerations for IP Fragment Filtering
- 2131 — BootP or DHCP Helper
- 2474 — DiffServ Definition
- 2475 — DiffServ Architecture
- 2578 — Structure of Management Information Version 2 (SMIv2)
- 2597— Assured Forwarding PHB Group
- 2615 — PPP over Sonet or SDH
- 2784 — Generic Routing Encapsulation (GRE) (partially supported)
- 3246 — An Expedited Forwarding PHB
- 3768 — VRRP
- 4447 — Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)
- 4459 — MTU and Fragmentation - Support fragment of inner packet (partially supported)
- 768 — UDP
- 783 — TFTP
- 791 — IP
- 792 — ICMP
- 793 — TCP
- 826 — ARP
- 894 — IP over Ethernet
- 903 — RARP
- 906 — TFTP Bootstrap
- 950 — Subnets
- 951 — BootP

RFC compliance - management

- 854 — TELNET
- 1213 — MIB II

  Refer to the *IronWare MIB Reference* for details.
- 1445 — Administrative Model for SNMPv2 - Support for View Subtree (partially supported)
- 1492 —TACACS+
- 1724 — RIPv2 MIB
- 1757 — RMON Groups 1, 2, 3, 9

  Refer to the *IronWare MIB Reference* for details.
- 2030 — SNTP
- 2068 — HTTP
- 2284 — PPP EAP -Support EAP extension
- 2578 — SNMPV2

- 2579 —Textual Conventions for SMIv2
- 2665 — Ethernet-Like MIB
- 2674— 802.1Q and 802.1p Bridge MIB
- 2787— VRRP-MIB
- 2863 — Interfaces Group MIB
- 2865 — RADIUS
- 2866 — RADIUS Accounting
- 2868 — RADIUS Attributes for Tunnel Protocol (partially supported)
- 2869 — RADIUS Extensions - EAP Message (type 79) and Message-Authenticator (type 80)
- 2933 — IGMP MIB
- 2934 — PIM MIB
- 3164 — BSD Syslog Protocol
- 3176 — InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched And Routed Networks.
- 3410 — SNMPV3
- 3411— Architecture for SNMP
- 3412 — Message Processing and Dispatching for SNMP
- 3413 — Simple Network Management Protocol (SNMP) Applications (partially supported)
- 3414 — USM for SNMPV3
- 3415 — VACM for SNMPV3
- 3416 — Version 2 of the Protocol Operations for the SNMP
- 3418 — Management Information Base (MIB) for the SNMP
- 3579 — RADIUS Support for Extensible Authentication Protocol (EAP) (partially supported)
- 3584 — Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework
- 3592 — SDH or SONET MIB (partially supported)
- 3812 — MPLS TE Standard MIB
- 3815 — Managed Objects for the Multiprotocol Label Switching (MPLS) and Label Distribution Protocol (LDP)

  Refer to the *IronWare MIB Reference* for details.
- 3826 — The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model
- 4188 — Definitions of Managed Objects for Bridges
- 4251 — The Secure Shell (SSH) Protocol Architecture
- 4252 — The Secure Shell (SSH) Authentication Protocol
- 4253 — The Secure Shell (SSH) Transport Protocol
- 4254 — The Secure Shell (SSH) Connection Protocol
- 4273 — Definitions of Managed Objects for BGP-4
- 4382 — MPLS or BGP Layer 3 Virtual Private Network (VPN) Management Information Base
- 4444 — Management Information Base for Intermediate System to Intermediate System (IS-IS)

- draft-grant-tacacs-02.txt — The TACACS+ Protocol
- draft-ietf-bfd-mib — Bidirectional Forwarding Detection Management Information Base
- draft-ietf-pwe3-enet-mib-11.txt — Ethernet Pseudo Wire (PW) Management Information Base
- draft-ietf-pwe3-pw-mib-11.txt — PW-STD-MIB Definitions (read-only)

RFC compliance - IPv6 management

- 2466 — IPv6 MIB for ICMPv6 Group

RFC compliance - IPv6 core

- 1191— Path MTU Discovery (partially supported)
- 1887 — IPv6 Unicast Address Allocation Architecture
- 1981 — Path maximum transmission unit (MTU) discovery for IPv6
- 2374 — IPv6 aggregatable global unicast address format
- 2375 — IPv6 Multicast Address Assignments
- 2450 — Proposed TLA and NLA Assignment Rules
- 2460 — IPv6 Specification
- 2461 — IPv6 Neighbor Discovery
- 2462 — IPv6 Stateless Address Auto-configuration
- 2463 — ICMPv6 (superseded by RFC 4443)
- 2464 — Transmission of IPv6 over Ethernet Networks
- 2471 — IPv6 Testing Address Allocation
- 2526 — Reserved IPv6 Subnet Anycast Address
- 2711 — IPv6 Router Alert Option
- 2928 — Initial IPv6 subTLA ID Assignments
- 3513 — IPv6 addressing architecture
- 3587 — IPv6 Global Unicast Address Format
- 3596 — DNS support
- 4007 — IPv6 scoped address architecture
- 4193 — Unique Local IPv6 Unicast Addresses
- 4291 — IPv6 Addressing architecture
- 4443 — ICMPv6 (supersedes RFC 2463)

RFC compliance - IPv6 routing

- 2080 — RIPng for IPv6
- 2472 — IPv6 over PPP
- 2545 — Use of MP-BGP-4 for IPv6
- 2740 — OSPFv3 for IPv6
- 5095 — Deprecation of Type 0 Routing Headers in IPv6

- draft-ietf-isis-igp-p2p-over-lan

RFC compliance - IPv6 multicast

- 2710 — Multicast Listener Discovery (MLD) for IPv6
- 3810 — Multicast Listener Discovery Version 2 for IPv6
- 4604 — IGMPv3 & MLDv2 for SSM

RFC compliance - IPv6 transitioning

- 2893 — Transition Mechanisms for IPv6 Hosts and Routers
- 3056 — Connection of IPv6 Domains through IPv4 Clouds

RFC compliance - MPLS

- 2205 — RSVP v1 Functional Specification
- 2209 — RSVP v1 Message Processing Rules
- 2702 — TE over MPLS
- 2747 — RSVP Cryptographic Authentication
- 3031 — MPLS Architecture
- 3032 — MPLS Label Stack Encoding
- 3036 — LDP Specification
- 3037— LDP Applicability
- 3209 — RSVP-TE (partially supported)
- 3270 — MPLS Support of Differentiated Services (partially supported)
- 3784 — ISIS-TE
- 4090 — Fast Re-Route for RSVP-TE Extensions (partially supported)
- 4448 — Encapsulation Methods for Transport of Ethernet over MPLS Networks

RFC compliance - L3VPN

- 2858 — Multiprotocol Extensions for BGP-4
- 3107 — Carrying Label Information in BGP-4
- 4364 — BGP or MPLS IP VPNs
- 4365 — Applicability Statement for BGP or MPLS IP VPNs
- 4382 — MPLS or BGP Layer 3 VPN MIB
- 4576 — Using LSA Options Bit to Prevent Looping in BGP or MPLS IP VPNs (DN Bit)
- 4577 — OSPF as the PE or CE Protocol in BGP or MPLS IP VPNs
- 4762 — Virtual Private LAN Service (VPLS) Using LDP Signaling (partially supported)
- draft-ietf-l2vpn-l2-framework — Framework for Layer 2 Virtual Private Networks

Internet drafts

In addition to the RFCs listed in "RFC COMPLIANCE", the PowerConnect supports the following Internet drafts:

- Draft-ietf-tcpm-tcpsecure-TCP Security
- Draft-ietf-isis-IPv6 for IPv6 with IS-IS
- Draft-ietf-idr-restart - Graceful Restart Mechanism for BGP
- Draft-ietf-idr-route-filter- Cooperative Route Filtering Capability for BGP-4
- Draft-ietf-ssm-arch SSM for IP
- Draft-ietf-pim-sm-v2-new - PIM-SM Protocol Specification, SSM Mode of Operation (partially supported)
- Draft-ietf-isis-igp-p2p-over-lan - Point-to-point operation over LAN in link state routing protocols (partially supported)
- Draft-ietf-bfd-base.txt - BFD
- Draft-ietf-bfd-v4v6-1hop.txt - BFD for IPv4 and IPv6 (Single Hop)
- Draft-ietf-bfd-generic.txt - Generic Application of BFD
- Draft-ietf-idr-bgp-ext-communities - BGP Extended Communities Attribute
- Draft-ietf-l3vpn-mpls-vpn-mib - MPLS or BGP Layer 3 VPN MIB
- Draft-ietf-l2vpn-requirements - Service Requirements for Layer 2 Provider Provisioned Virtual Private Networks
- draft-ietf-l2vpn-vpls-mib-01, VPLS-Generic-Draft-01-Mib module. No other module of this draft is supported.
- Draft-ietf-pwe3-arch - PWE3 Architecture
- Draft-ietf-pwe3-ethernet-encap - Encapsulation Methods for Transport of Ethernet Frames Over IP or MPLS Networks
- Draft-ietf-pwe3-pw-tc-mib - Definitions for Textual Conventions and OBJECT-IDENTITIES for Pseudo-Wires Management
- Draft-ietf-pwe3-pw-mib - Pseudo Wire (PW) Management Information Base

NOTE
For standards that are partially supported, contact Dell for more details.

Appendix

Acknowledgements

E

This appendix presents the acknowledgements for portions of code from various vendors that are included in the PowerConnect devices covered in this manual.

OpenSSL license

Copyright (c) 1998-2001 The OpenSSL Project. All rights reserved.

1. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
2. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
3. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation or other materials provided with the distribution.
4. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (*http://www.openssl.org/*)”
5. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
6. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
7. Redistributions of any form whatsoever must retain the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (*http://www.openssl.org/*)”

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Cryptographic software

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)/. All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

1. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
2. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
3. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation or other materials provided with the distribution.
4. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).
5. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:

   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]