53-1001810-01
January 31, 2010

BigIron RX

Configuration Guide

Supporting Multi-Service IronWare v02.7.02

53-1001810-01

Notes, Cautions, and Warnings

NOTE
A NOTE indicates important information that helps you make better use of your computer.

CAUTION
See the safety and regulatory information that shipped with your system. For additional regulatory information, see the Regulatory Compliance Homepage on www.dell.com at the following location: www.dell.com/regulatory_compliance.

CAUTION
A CAUTION indicates potential damage to hardware or loss of data if instructions are not followed.

DANGER
A DANGER indicates a potential for property damage, personal injury, or death.

Regulatory Model Codes: Brocade DCX-4S, Brocade DCX

Contents

Chapter 7 Configuring IP

Chapter 15 Virtual Switch Redundancy Protocol (VSRP)

Chapter 16 Topology Groups

Chapter 17 Configuring VRRP and VRRPE

Chapter 18 Configuring Quality of Service

About This Document

In this chapter

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 Brocade Layer 3 Switch, you should be familiar with the following protocols if applicable to your network – IP, RIP, OSPF, BGP, ISIS, IGMP, PIM, DVMRP, and VRRP.

Supported hardware and software

Although many different software and hardware configurations are tested and supported by Brocade Communications Systems, Inc., documenting all possible configurations and scenarios is beyond the scope of this document.

This guide presents features in software release 02.7.02.

The following hardware platforms are supported by the 02.7.02 software release of this guide:

- BigIron RX - 4
- BigIron RX - 8
- BigIron RX - 16
- BigIron RX - 32

Features that are not documented in this guide are not supported.

List of supported features

Features or options not listed in the Supported and unsupported features table or documented in this guide are not supported.

TABLE 1 Supported and unsupported features

Category	Feature description
System level features	
Cisco Discovery Protocol (CDP)	Allows you to configure a Brocade 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.
CLI Logging	
Denial of Service (DoS) protection	Protection from SYN attacks Protection from Smurf attacks
Foundry Discovery Protocol (FDP)	Enables Brocade devices to advertise themselves to other Brocade devices on the network.
High Availability	OS Layer 2 Hitless Software Upgrade
Management Options	Serial and Telnet access to industry-standard Command Line Interface (CLI) SSHv2 TFTP Web-based GUI SNMP versions 1, 2, and 3 *IronView Network Manager* .
Security	AAA Authentication Local passwords RADIUS Secure Shell (SSH) version 2 Secure Copy (SCP) TACACS/TACACS+ User accounts 802.1x: All EAP types, including MD5, TLS, TTLS, and PEAP Multi-device port authentication AES for SNMPv3, SSHv2, SCP, and HTTPS Note:Telnet, SSH, Web and SNMP servers are disabled by default, and can be enabled selectively.
CPU protection	There are no CLI commands for CPU protection. The device forwards unknown unicast, broadcast and multicast packets in hardware; therefore, the CPU is automatically 'protected' from having to handle too many packets.
SysLogD Server Logging	Multiple SysLogD server logging
sFlow	sFLow version 5
Uni-directional Link Detection (UDLD)	Monitors a link between two Brocade devices and brings the ports on both ends of the link down if the link goes down at any point between the two devices.
Layer 2 features	

TABLE 1 Supported and unsupported features (Continued)

Category	Feature description
802.1d	Spanning Tree Protocol (STP) and Single Spanning Tree Protocol (SSTP)
802.1p	Quality of Service (QoS) queue mapping
802.1q	See VLANs, below
802.1s	Multiple Spanning Tree Protocol (MSTP)
802.1w	Rapid Spanning Tree Protocol (RSTP)
802.3ad	Dynamic Link Aggregation on tagged and untagged trunks
Jumbo packets	Layer 2 jumbo packet support
Layer 2 Hitless failover	
Layer 2 IGMP Snooping	
L2 ACL	Filtering based on MAC layer-2 parameters.
MAC Filtering	MAC filtering and address-lock filters to enhance network security
MRP	Metro Ring Protocol (MRP) Phase 1 and 2
PVST / PVST+	Per-VLAN Spanning Tree (PVST)
Rate Limiting	Port-based, port-and-priority based, port-and-vlan-based, and port-and-ACL-based rate limiting on inbound ports are supported.
SuperSpan	A Brocade 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 • Brocade MRP • VSRP • 802.1W
Trunk Groups and LAG	Allows you to manually configure multiple high-speed load-sharing links between two Brocade devices or between a Brocade device and a server.
VLANs	802.1Q tagging Port-based VLANs Super Aggregated VLANs (SAV) Dual-mode VLAN ports Transparent Port Flooding VLAN ID to MSTP Instance Pre-assignment Private VLANs
VSRP	Layer 2 Virtual Switch Redundancy Protocol (VSRP) Layer 3 Virtual Switch Redundancy Protocol (VSRP) VSRP and MRP Signaling
Layer 2 ACLs	Replaces MAC filters
Layer 2 PIM Snooping	
Layer 3 features	

TABLE 1 Supported and unsupported features (Continued)

Category	Feature description
ACLs	Standard, Extended and Super Inbound ACL logging ACL editing
BGP	BGP routes BGP peers BGP dampening Graceful Restart
FDR	Foundry Direct Routing
IP Forwarding	IPv4 Routing IPv6 Routing
IP Static entries	Routes ARPs Virtual interfaces Secondary addresses
IS-IS	
Multicast Routing	Multicast cache L2 IGMP table DVMRP routes PIM-DM PIM-SM PIM-SSM PIM Snooping
OSPF	OSPF routes OSPF adjacencies - Dynamic OFPF LSAs OSPF filtering of advertised routes
PBR	Policy Based Routing (Release 02.2.01 and later)
RIP versions 1 and 2	RIP routes
VRRP and VRRPE	Virtual Router Redundancy Protocol (VRRP) and VRRP Extended (VRRPE)
IPv6 features	
IPv6 ACLs	Extended ACLs
IPv6 Routing Protocols	RIPng OSPFv3 BGP4+

Unsupported features

The following features are not supported in software release 02.7.00 on device:

- AppleTalk
- Dynamic IP Routing

- IPX
- Mirroring across VLANs
- MPLS
- NAT
- RARP
- VLANs
  - VLAN translation
  - Subnet VLANs
- Source IP Port Security

What's new in this document

Enhancements and configuration notes in release 02.7.02

The following table provides a brief description of the enhancements added in this release and a reference to the specific chapter, and section in the *BigIron RX Configuration Guide* or the *Brocade BigIron RX Series Installation Guide* that contain a detailed description and operational details for the enhancement.

TABLE 2 Summary of enhancements in release 02.7.02

Enhancement	Description	See page
System features		
Enhanced **spreed-duplex** command	The **speed-duplex** command has been enhanced to support 24F and 24HF modules. The **auto (**Autonegotiation mode) option has also been added to allow the user to set the speed on E1MG-TX media.	**Book:** *BigIron RX Configuration GiuideBigIron RX Series Configuration Guide* **Chapter:** "Configuring Interface Parameters" **Section**: "Speed/Duplex negotiation"

Enhancements and configuration notes in release 02.7.01

TABLE 3 Summary of enhancements in release 02.7.01

Enhancement	Description	See page
System features		
New 16x10G module.iew	The new 16 port 10GE oversubscribed module provides 4:1 over-subscription on the network ports. The new module is compatible with all previous modules on the BigIron RX.	**Book:** *Brocade BigIron RX Series Installation Guide* **Chapter:** Product Overview **Section**: 16-port 10 Gigabit Ethernet Oversubscribed Module **Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Quality of Service" **Section**: "QoS for the oversubscribed 16 x 10GE modules"
Network management		
128-bit AES encryption support for SNMP V3	The Advanced Encryption Standard (AES) provides one of the most advanced encryption capabilities available today. This release adds AES for SNMPv3 as specified in RFC 3826. To enable AES encryption, specify the **aes** encryption type when defining an SNMP user account.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Securing SNMP Access" **Section**: "Defining an SNMP user account"
AES Encryption for SSH v2, Secure Copy (SCP), and Secure HTTPS (HTTPS)	SSH v2, SCP, and HTTPS now supports a very strong AES encryption algorithm in the following modes: aes256-cbc, aes192-cbc, and aes128-cbc.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Securing SNMP Access" **Section**:

Enhancements and configuration notes in release 02.7.00

TABLE 4 Summary of enhancements in release 02.7.00

Enhancement	Description	See page
Layer 1 features		
New Optics Support	The SFP-compliant E1MG-TX fiber-optic module now supports speeds of 10/100/1000.	**Book:** *Brocade BigIron RX Series Installation Guide*
UDLD Start-up Mode	In this release, after UDLD is enabled on a port, UDLD can be configured to be kept in a newly created suspended state until it receives its first keep-alive message from the other end.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring Uni-Directional Link Detection (UDLD)"

TABLE 4 Summary of enhancements in release 02.7.00 (Continued)

Enhancement	Description	See page
Multicast, Broadcast, and Unknown Unicast Rate Limiting per Module	This release introduces a new hardware (module) based Multicast/Broadcast/Unknown Unicast Rate-Limiting for both CPU based flooding and Hardware based flooding.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Traffic Reduction" **Section**: "NP based multicast, broadcast, and unknown-unicast rate limiting"
Link Layer Discovery Protocol (LLDP)	Beginning with release 02.7.00, Link Layer Discovery Protocol (LLDP) is supported. This protocol enables a station to advertise its capabilities to, and to discover other LLDP-enabled stations in the same 802.1AB LAN segments.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring LLDP"
CLI Change	To globally enable MAC port security, the **global-port-security** command has been added. The **port security** command is now only used when configuring MAC port security on specific interfaces.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Using the MAC Port Security Feature" **Section**: "Enabling the MAC port security feature"
Network management		
DHCP Relay Enhancement	Beginning with this release, the IP subnet configured on the port which is directly connected to the device sending a BootP/DHCP request, does not have to match the subnet of the IP address given by the DHCP server.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:** "Configuring BootP/DHCP forwarding parameters"
SNMP MIBs for Layer 2 ACLs and Filters	The following MIB tables have been added to this release: • Textual Conventions • Layer 2 ACL Next Clause Table • Layer 2 ACL Configuration Table • Layer 2 ACL Binding Configuration Table	**Book:** *MIB Reference* **Chapter:** Filtering Traffic **Section:** Layer 2 ACLs

Enhancements and configuration notes in release 02.6.00

TABLE 5 Summary of enhancements in release 02.6.00

Enhancement	Description	See page
Layer 1 features		
Digital Optical Monitoring	Beginning with release 0 2.6.00, Digital Optical Monitoring will only support newly qualified 1Gigabit optics. Digital Optical Monitoring for previous 1Gigabit optics that do not include "OM" after the model numbers will not be able to use this feature.	**Book:** *Brocade BigIron RX Series Installation Guide* **Chapter**: Connecting a device Switch to a Network Device **Section**:Digital Optical Monitoring
BFD for IS-IS, OSPFv2 and OSPF v3	device provides support for Bidirectional Forwarding Detection (BFD) in Version 02.6.00 of the Multi-Service IronWare software.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "BiDirectional Forwarding Detection (BFD)"
LACP Continous Fast Timer	In a dynamic or keep-alive LAG, a port's timeout can be configured as short or long	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Link Aggregation" **Section:** "Configuring an LACP timeout"
Rate Limiting ARP Packets	This new feature allows you to rate-limit ARP traffic that is destined for CPU of the device router.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:**"Applying a rate limit to ARP packets on an interface"
Layer 2 features		
VSRP Fast Start	Non-Brocade or non-VSRP aware devices connected to a VSRP master can now quickly switch over to the new master when a VSRP failover occurs.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Virtual Switch Redundancy Protocol (VSRP)" **Section:** "VSRP fast start"
LACP Enhancements	Beginning with release 02.6.00 of the Multi-Service IronWare software, all trunking and link aggegation configuration has been revamped and placed under a single interface. This new interface allows you to configure either of the previously supported LAG types: Static LAGs and Dynamic LAGs as well as the new "Keep Alive" LAGs. The new LAG configuration procedures supersede the previous configurations procedures for Trunks and Dynamic Link Aggregation.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Link Aggregation"

TABLE 5 Summary of enhancements in release 02.6.00 (Continued)

Enhancement	Description	See page
Multicast Layer 2 Filter	Beginning with release 02.6.00, you can define multicast boundaries on a per VLAN basis.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Protocols" **Section:** "Layer 2 multicast filters"
Layer 3 features		
IPv6 PIM-SM	In Release 02.6.00 of the Multi-Service IronWare software, the device supports IPv6 Protocol Independent Multicast (PIM) Sparse. IPv6 PIM Sparse provides multicasting that is especially suitable for widely distributed multicast environments	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IPv6 Multicast Features" **Section:** "IPv6 PIM-sparse mode"
IPv6 Embedded RP	This release supports Embedded RP which allows the switch to learn RP information using the multicast group destination address instead of the statically configured RP.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IPv6 Multicast Features" **Section:** "Embedded Rendezvous Point (RP)"
IPv4 PIM Snooping	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	**Book:** *BigIron RX Series Configuration Guide* **Chapter**: "Configuring IP Multicast Protocols" **Section:** "PIM SM traffic snooping"
Anycast RP	This release supports Anycast RP as defined in RFC 3446. 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.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Protocols" **Section:** "Anycast RP"
Multicast Listening Discovery (MLD)	Release 02.6.00 adds support for MLD Snooping (MLDv1 and MLDv2) on Brocade BigIron RX devices running IPv6.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IPv6 Multicast Features" **Section:** "Multicast Listener Discovery and source specific multicast protocols(MLDv2)"
IGMPv3 and IGMP Snooping	In Release 02.6.00 of the Multi-Service IronWare software, creating an IGMP static-group allows the BigIron RX switch having L2 interfaces configured with snooping to pull traffic from upstream sources using IGMP joins. When using the uplink option, you avoid burning a dedicated port. This is supported for IGMP v2 and v3.	**Book:** *BigIron RX Series Configuration Guide* **Chapter**: "Configuring IP Multicast Protocols" **Section**: "IGMP v3"
IGMP v3 Static Client	In Release 02.6.00 of the Multi-Service IronWare software, creating an IGMP static-group allows the BigIron RX switch having L2 interfaces configured with snooping to pull traffic from upstream sources using IGMP joins. When using the uplink option, you avoid burning a dedicated port. This is supported for IGMP v2 and v3.	**Book:** *BigIron RX Series Configuration Guide* **Chapter**: "Configuring IP Multicast Protocols" **Section:** "Creating a static IGMP group"

TABLE 5 Summary of enhancements in release 02.6.00 (Continued)

Enhancement	Description	See page
IGMP v3 Fast Leave and Tracking	In Release 02.6.00 of the Multi-Service IronWare software, you can configure a device running IGMP Snooping to immediately remove a VLAN from the IP multicast group when it detects a fast leave message on a specified VLAN.	**Book:** *BigIron RX Series Configuration Guide* **Chapter**: "Configuring IP Multicast Protocols" **Section:**"Enabling membership tracking and fast leave"
Static Route ARP Validate Next Hop	Beginning with release 02.6.00, you can configure the BigIron RX to perform multicast validation checks on the destination MAC address, the sender and target IP addresses, and the source MAC address.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Protocols" **Section:** "Next hop validation check"
IGMP Proxy per VLAN or instance	Introduced in version 02.6.00 of the Multi-Service IronWare software, multicast traffic can be reduced by configuring an BigIronRX switch to issue IGMP host 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 IGMP tasks upstream of the discovered hosts. Where there are multiple IGMP hosts downstream, this removes the need to send multiple messages.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Traffic Reduction" **Section:** "Multicast traffic reduction per VLAN"
Layer 4 features		
Automatic ACL Rebind	Beginning wirh release 02.6.00, the ACL automatic rebind feature allows the newly changed ACL filter definitions to be automatically applied to the ports where the ACL was bound.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Access Control List" **Section:** "ACL automatic rebind"
Network management		
Support for BFD MIB and SNMP Traps	Support for BFD IETF draft mib version 3 (draft-ietf-bfdmib-03.mib) with this release as described in the *Management Information Base Reference.*	**Book:** *MIB Reference* **Chapter:** Bidirectional Forwarding

Enhancements and configuration notes in patch release 02.5.00c

TABLE 6 Summary of enhancements in release 02.5.00c

Enhancement	Description	See page
Super ACLs	With this patch release, the Multi-Service IronWare software supports Super ACLs that can match on fields in a Layer 2 or Layer 4 packet header.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Access Control List" **Section:** "Configuring super ACLs"

Enhancements and configuration notes in patch release 02.5.00b

TABLE 7 Summary of enhancements in release 02.5.00b

Enhancement	Description	See page
ACL-based Inbound sFlow	With this patch release, the Multi-Service IronWare software supports using an IPv4 ACL to select packets that should be collected as special sFlow samples, in addition to the regular statistical sampling of sFlow.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "sFlow" **Section:** "ACL-based inbound sFlow"

Summary of enhancements and configuration notes in release 02.5.00

TABLE 8 Summary of enhancements in release 02.5.00

Enhancement	Description	See page
BigIron RX-32 Chassis	Release 02.5.00 introduces the BigIron RX-32 chassis which runs the same Multi-Service IronWare software as other chassis in the BigIron RX series. The new BigIron RX-32 chassis provide support for up to 32 interface modules.	**Book:** *Brocade BigIron RX Series Installation Guide*
New Process for Upgrading Multi-Service IronWare Software	The software images required for operating the device switch remain the same however, beginning with version 02.5.00 of the Multi-Service IronWare software, the upgrading procedures have been changed. The new procedure is described in the Release Notes for device – Multi-Service IronWare Software Release 02.5.00.	**Book:** Release Notes for device – Multi-Service IronWare Software Release 02.5.00.
SDS Over Telnet	Beginning with release 02.5.00 of the Multi-Service IronWare software, remote SDS is supported. This feature will dramatically improve the ability to troubleshoot issues on the line-card without the need of a serial cable.	N/A
Enhancement on Static ARP	In Release 02.5.00 of the Multi-Service IronWare software, static ARP has been enhanced to support the ability to create a static ARP entry without an outgoing interface.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:** "Creating a floating static ARP entry"
Static Route ARP Validate Next Hop	Beginning with release 02.5.00, you can configure the BigIron RX to perform validation checks on the destination MAC address, the sender and target IP addresses, and the source MAC address.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:** "Static route ARP validation check"
Multicast MII Sharing	In Release 02.5.00, the multicast hardware device drivers have been enhanced to optimize utilization and improve overall performance.	N/A

TABLE 8 Summary of enhancements in release 02.5.00 (Continued)

Enhancement	Description	See page
Multicast	Starting release 02.5.00, low priority multicast traffic is rate-limited to 1.8 Gbps per packet processor.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Quality of Service" **Section:** "Configuring multicast traffic engineering"
Changes to the **copy tftp Image** command	In Release 02.5.00 of the Multi-Service IronWare software, new option have been added to the **copy tftp image** command to enable the user to upgrade the boot, monitor, and MBRIDGE only when needed.	**Book:** Release Notes for device – Multi-Service IronWare Software Release 02.5.00.
New MIB Objects	The following MIB objects have been added to the snIfStpTable: • snIfStpPortRole • snIfStpBPDUTransmitted • snIfStpBPDUReceived • snIfRstpConfigBPDUReceived • snIfRstpTCNBPDUReceived • snIfRstpConfigBPDUTransmitted • snIfRstpTCNBPDUTransmitted	**Book:** *MIB Reference* **Chapter:** Interfaces **Section:** Port STP Configuration Groups

Summary of enhancements and configuration notes in patch release 02.4.00c

TABLE 9 Summary of enhancements in release 02.4.00c

Enhancement	Description	See page
ACL Based RP assignment	The **rp-address** command has been enhanced to allow multiple static RP configurations.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Protocols" **Section:** "ACL based RP assignment"
Route Selection Precedence for Multicast	In patch 02.4.00c, the **route-precedence** command allows the user to specify a precedence table that dictates how routes are selected for multicast.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Protocols" **Section:**"Route selection precedence for multicast"

Summary of enhancements and configuration notes in release 02.4.00

TABLE 10 Summary of enhancements in release 02.4.00

Enhancement	Description	See page
US Daylight Saving Time scheme	The new Daylight Saving Time (DST) change that went into effect on March 11th, 2007 affects only networks following the US time zones. However, to trigger the device to the correct time, the device must be configured to the US time zone, not the GMT offset.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Basic Parameters" **Section:** "New Daylight Saving Time (DST)"
New **show boot-image** command	Using the **show boot-image** command displays which image the device will use for the next reboot or reload.	**Book:** *Brocade BigIron RX Series Installation Guide* **Chapter:** Upgrading Software Images and Configuration Files **Section:** Displaying the Next Boot Image
New **show image_checksum** command	The **image_checksum** command will allow the user to verify the checksum of a image.	**Book:** *Brocade BigIron RX Series Installation Guide* **Chapter:** Upgrading Software Images and Configuration Files **Section:** Verifying the Checksum of an Image
Private VLAN	A private VLAN is a VLAN that has the properties of standard Layer 2 port-based VLANs but also provides additional control over flooding packets on a VLAN.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"VLANs" **Section:** "Private VLANs"
MRP Phase 2	In Metro Ring Protocol (MRP) Phase 2, the same physical interface can be shared by multiple rings belonging to the same VLAN.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Metro Ring Protocol (MRP) Phase 1 and 2" **Section:** "MRP phase 2"
Outbound Rate Limiting	Outbound rate limiting support has been added to this release.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Traffic Reduction"
Increase Global Static ARP Entries	The system max value for ip-static-arp can be configured to values up to 16,384 beginning with version 02.4.00 of the device Multi-Service IronWare software.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:** "Changing the maximum transmission unit on an individual interface"
OSPF ABR Type 3 LSA Filtering	The OSPF ABR Type 3 LSA Filtering feature extends the ability of an ABR that is running the OSPF protocol to filter type 3 link-state advertisements (LSAs) that are sent between different OSPF areas.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring OSPF Version 2 (IPv4)" **Section:** "OSPF ABR type 3 LSA filtering"

TABLE 10 Summary of enhancements in release 02.4.00 (Continued)

Enhancement	Description	See page
New show OSPF neighbor by area command	This feature allows OSPF to display the OSPF neighbors existing in a particular area.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring OSPF Version 2 (IPv4)" **Section:** "Displaying OSPF neighbor information"
Track IP route time in show command	The **show ip route** command has been enhanced to include the elapse time since an IP route was installed.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:** "Displaying the IP route table"
Compare MED for internal BGP route with empty as-path	This new BGP command directs iBGP to take the MED value into consideration even if the route has an empty as-path path attribute.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring BGP4 (IPv4 and IPv6)" **Section:** "Configuring the BigIron RX to always compare Multi-Exit Discriminators (MEDs)"
OSPF Default Network Route	This feature enables the device to use default route (0.0.0.0/0) to a resolve static OSPF route. Note:This differs from the default behavior in previous versions of Multi-Service IronWare software.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring OSPF Version 2 (IPv4)" **Section:** "Configuring a default network route"
IPv6 Default Route ECMP	This feature allows for load distribution of traffic among the available IPv6 default route next-hops.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Basic IPv6 Connectivity" **Section:** "ECMP load sharing for IPv6"
IPv6 Tunneling in Hardware	Manual configuration of IPv6 to IPv4 tunnels is now supported in this release. These tunnels will be installed into the hardware route table and tunnel encapsulation and decapsulation is done in hardware.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:**"IPv6 over IPv4 tunnels in hardware"
IPv6 Load Sharing over ECMP and Trunks	When the device receives traffic for a destination, and the IPv6 route table contains multiple, equal-cost paths to that destination, the packets are load balanced between multiple next-hops including member ports of a trunk.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Basic IPv6 Connectivity" **Section:** "ECMP load sharing for IPv6"
Directly Attached Host Resource Allocation	The CAM allocations can be re-distributed using the **cam-partition next-hop** command.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Basic Parameters" **Section:** "Re-distributing CAM allocations"

TABLE 10 Summary of enhancements in release 02.4.00 (Continued)

Enhancement	Description	See page
Multicast Boundaries	The Multicast Boundary feature is designed to selectively allow or disallow multicast flows to configured interfaces.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Protocols" **Section:** "IP multicast boundaries"
MBGP for IPv6	This release supports the Multi-protocol Border Gateway Protocol (MBGP) for IPv6.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IPv6 MBGP"
IPv6 mroute	This release supports multicast route table ipv6 multicast display and management.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IPv6 Routes" **Section:** "Configuring a IPv6 multicast route"
Passive Multicast Route Insertion (PMRI)	This new feature prevents unwanted multicast traffic from being sent the CPU by conditionally dropping unwanted multicast traffic in hardware.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Protocols" **Section:** "Passive Multicast Route Insertion (PMRI)"
IP Source Guard	IP source guard is used on client ports to prevent IP source address spoofing.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Inspecting and Tracking DHCP Packets." **Section:** "IP source guard"
Dynamic ARP Inspection	Dynamic ARP Inspection (DAI) is a security feature that can prevent Man-in-the-Middle (MiM) or ARP spoofing/poisoning attacks.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Inspecting and Tracking DHCP Packets" **Section:** "Dynamic ARP inspection"
DHCP Snooping with Option 82	This feature allows the device to snoop DHCP packets for Dynamic ARP inspection and allows for the insertion of DHCP Option 82 attributes into the DHCP packet prior to relaying to the DHCP server.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Inspecting and Tracking DHCP Packets" **Section:** "DHCP relay agent information (DHCP option 82)"
DoS Protection	This feature allows for monitoring the hit rate of the ACL and drops matching traffic above a selected rate and locking the port if the rate exceeds a maximum allowed amount.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Protecting Against Denial of Service Attacks" **Section:**"ACL-based DOS-attack prevention"

TABLE 10 Summary of enhancements in release 02.4.00 (Continued)

Enhancement	Description	See page
ACL-Based Mirroring	With this release, the Multi-Service IronWare software supports using an ACL to select traffic for mirroring from one port to another.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Access Control List" **Section:** "ACL-based inbound mirroring"
ip dns domain-list command	This feature is designed to define a list od domain names that are used in order to resolve a host.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:** "Defining a DNS entry"
CLI Logging	This feature provides the logging of all valid CLI commands from each user session into the system log.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Using Syslog" **Section:**"Logging all CLI commands to Syslog"
Syslog Source Interface	You can configure the BigIron RX 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.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:**"Configuring an interface as the source for Syslog packets"
UDLD Traps and Syslogs	UDLD state changes will now be logged by default.	**Book:** *MIB Reference* **Chapter:** Traps and Objects to Enable Traps **Section:** UDLD Traps
New Brocade MIB objects	The following MIBs have been depreciated by snAgentCpuUtilTable: • snAgGblCpuUtil1SecAvg • snAgGblCpuUtil5SecAvg • snAgGblCpuUtil1MinAvg	**Book:** *MIB Reference* **Chapter:** Monitoring and Logging **Section:** Usage Notes on CPU Utilization and System CPU Utility Table

Summary of enhancements in patch release 02.3.00a

TABLE 11 Summary of enhancements in patch release 02.3.00a

Enhancement	Description	See...
Transparent Port Flooding	When the Transparent Port Flooding feature in enabled for a port, all MAC learning will be disabled for that port. This will result in all Layer 2 traffic to be flooded to all other ports within the VLAN. Starting with release 02.3.00a.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Using the MAC Port Security Feature" **Section:**"Transparent port flooding"
VLAN ID to MSTP Instance Pre-assignment	This feature will allow the user to assign a VLAN ID to a Common Spanning Tree (CIST), or Multiple Spanning Tree Instance (MSTI) even though a VLAN has not been created yet. Starting with release 02.3.00a.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Multiple Spanning Tree Protocol (MSTP) 802.1s" and "VLANs" **Section:**"Configuring an MSTP instance"

Summary of enhancements and configuration notes in release 02.3.00

System enhancements

TABLE 12 System enhancements

Enhancement	Description	See...
New Hardware Support	The following new hardware is supported with the 02.3.00 software release for the device: 1 **10G-XFP-CX4** - part number 10G-XFP-CX4 , A new XFP Module is available for use in the BigIron RX Series and 10G Interface Modules with the following capabilities: • 10GBASE-CX4 compliant per 802.1ak • CX4 connector • Up to 15 meter reach when using CX4 grade copper cables • Restriction of Hazardous Substances (RoHS) 5/6 compliant • Hot pluggable • Compatible with industry-standard MDI socket for CX4 • Supports 4 channel full-duplex copper cable 2 **10GBase-ZR** – part number 10G-XFP-ZR supports 1550 nm wavelength with a maximum distance of up to 80 km over single mode fiber (SMF). 3 **10GBase-ZRD** – part number 10G-XFP-ZRD supports 40 different wavelengths at 1550 nm. 4 **48-port 1 Gbps** Copper Ethernet interface module	**Book:** *Brocade BigIron RX Series Installation Guide*
Hitless OS Upgrade for Layer 2	Version 02.5.00 of the Multi-Service IronWare software supports hitless upgrade of the operating system on a device switch. Using this feature, you can upgrade the Multi-Service IronWare software without a loss or disruption of service as described.	**Book:** *Brocade BigIron RX Series Installation Guide* **Chapter:** Upgrading Software Images and Configuration Files **Section:**Layer 2 Hitless OS Upgrade
Logging of packets denied by ACLs.	You can restrict the number of times a message is logged in the Syslog due to packets that matches a deny ACL condition.	**Book**: *BigIron RX Series Configuration Guide* **Chapter:** "Access Control List" **Section:** "Enabling the new logging method"
Modifying ACLs	You can modify ACL entries anywhere in an ACL.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Access Control List" **Section:** "Enabling the new logging method"
SFM FE Monitoring	In this release, the Switch Fabric Module monitoring has been enhanced. If the SFM fails, it generates a syslog that includes the status of individual fabric elements on the SFM modules.	**Book:** *Brocade BigIron RX Series Installation Guide*

TABLE 12 System enhancements (Continued)

Enhancement	Description	See...
Enhanced Digital Optical Monitoring	You can configure the BigIron RX to monitor XFPs and SFPs in the system either globally or by specified port.	**Book:** *Brocade BigIron RX Series Installation Guide* **Chapter:** Connecting a BigIron RX Series Switch to a Network Device **Section:** Enhanced Digital Optical Monitoring
Re-distributing CAM Allocations	In releases prior to 02.3.00, CAM partitioning was not configurable. Starting in BigIron RX software release 02.3.00, you can specify the CAM assigned to each of the CAM entry types globally.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Basic Parameters" **Section:** "Re-distributing CAM allocations"
Enhanced SFM (power-off) command	You can disable power to a specified switch fabric module and then reenable it.	**Book:** *Brocade BigIron RX Series Installation Guide* **Chapter:** Managing the BigIron RX Series Chassis and Modules **Section:** Disabling and Reenabling Power to the Switch Fabric Modules
Enhanced **speed-duplex** command	In this release, the **speed-duplex** command has been enhanced to include the master and slave parameters.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Interface Parameters" **Section:**"Speed/Duplex negotiation"

Layer 2 enhancements

TABLE 13 Layer 2 enhancements

Enhancement	Description	See...
Flow based MAC Learning	In this release, the **cpu-flooding unknown-unicast** command that disables hardware flooding of unknown unicast on every VLAN has been added. This will allow MAC learning only where necessary and at a system level to allow more than 16k MACs.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "VLANs" **Section:** "Flow based MAC learning"
VSRP Slow-Start	This feature allows for a hold down time before the backup returns ownership to the master after the link is seen.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Virtual Switch Redundancy Protocol (VSRP)" **Section:**"VSRP slow start"
802.1s Multiple Spanning Tree Protocol (MSTP)	With this release, you can configure multiple STP instances using MSTP protocol, as defined in IEEE 802.1s	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Multiple Spanning Tree Protocol (MSTP) 802.1s"

Layer 3 enhancements

TABLE 14 Layer 3 enhancements

Enhancement	Description	See...
OSPF NBMA	You can configure an interface to send OSPF unicast packets rather than broadcast packets to its neighbor by configuring non-broadcast multi-access (NBMA) networks.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring OSPF Version 2 (IPv4)" **Section:**"Configuring an OSPF non-broadcast interface"
Layer 3 VSRP	VSRP redundancy and sub-second failover for Layer 3 topologies is available in this release.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Virtual Switch Redundancy Protocol (VSRP)" **Section:**"Enabling Layer 3 VSRP"
VSRP Delay Link Events	This is a new VSRP command that will delay the sending of port "up"/"down" events.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Interface Parameters" **Section:**"Port transition hold timer"
IPv6 Hardware Forwarding	Forwarding for Layer 3 IP switching technology for the forwarding of IPv6 packets.See the "Configuring Basic IPv6 Connectivity" chapter of the *BigIron RX Series Configuration Guide*.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring Basic IPv6 Connectivity"
OSPF v3	IPv6 supports OSPF version 3 (OSPFv3), which functions similarly to OSPF version 2.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring OSPF Version 3"
BGP+	Brocade's 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.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring BGP4+"
RIPng	IPv6 RIP, known as ***Routing Information Protocol Next Generation*** or ***RIPng***, functions similarly to IPv4 RIP version 2. RIPng supports IPv6 addresses and prefixes.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring RIPng"
ACL Duplication Check	The **acl-duplication-check** command has been changed to **acl-duplication-check-disable**. With this command, software checking for duplicate ACL entries will be disabled after an upgrade.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Access Control List" **Section:**"Enabling ACL duplication check"
IPv6 ACLs	In this release you can use an IPv6 ACL to provide input to other features such as route maps and distribution lists.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "IPv6 Access Control Lists (ACLs)"

TABLE 14 Layer 3 enhancements (Continued)

Enhancement	Description	See...
Default Originate Route for BGP	In this release, if a default route is not present in the IP routing table, the user can configure a major route to be used for forwarding packets to all unknown destination. Starting with release 02.3.00a.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring BGP4 (IPv4 and IPv6)" **Section:**"Originating the default route"
Changes to BGP4 Path Selection for a Route	With this release of Multi-Service IronWare, the process by which BGP selects a path has changed. The following procedure replaces the procedure described in the *BigIron RX Series Configuration Guide*.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring BGP4 (IPv4 and IPv6)" **Section:**"How BGP4 selects a path for a route"
BGP **allowas-in** command	The **allowas-in** command has been added to this release to allow you to set a parameter that disables the BGP AS_PATH check function for routes learned from a specified location.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring BGP4 (IPv4 and IPv6)" **Section:**"Configuring a switch to allow routes with its own AS number"
Default Route ECMP	This feature allows for load distribution of traffic among the available default route next-hops.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP" **Section:**"Default route ECMP"
Transparent Firewall Mode	The Transparent Firewall mode feature allows users to insert a Firewall in front of their existing network without changing the statically defined IP addresses of their network-connected devices. This will allow the users to permit selected devices from a subnet to cross the firewall while access to other devices on the same subnet are denied.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "VLANs" **Section:**"Transparent firewall mode"

IP multicast enhancements

TABLE 15 IP multicast enhancements

Enhancement	Description	See...
MBGP	Multiprotocol BGP allows for the inclusion of information other than IPv4 routes through BGP packets is available in this release.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring MBGP"
Multicast Source Discover Protocol (MSDP)	This release supports the Multicast Source Discovery Protocol (MSDP). It is used by Protocol Independent Multicast (PIM) Sparse routers to exchange routing information for PIM Sparse multicast groups across PIM Sparse domains.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP Multicast Protocols" **Section:**"Configuring Multicast Source Discovery Protocol (MSDP)"

TABLE 15 IP multicast enhancements (Continued)

Enhancement	Description	See...
MSDP Mesh Groups	This release supports Multicast Source Discovery Protocol (MSDP) Mesh Groups. This feature allows you to connect several RPs to each other which reduces the forwarding of SA messages within a domain.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP Multicast Protocols" **Section:**"Configuring MSDP mesh group"
IGMP v3	IGMP v3 provides selective filtering of traffic based on traffic source.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP Multicast Protocols" **Section:**"IGMP v3"
PIM-SSM v4	PIM-SSM is a routing protocol used for source specific multicast groups and is used in conjunction with IGMPv3	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP Multicast Protocols" **Section:**"PIM-SSMv4"
IGMP v2/v3 Fast Leave	IGMP Fast leave allows clients to leave groups without the three second waiting period, if certain conditions are met.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP Multicast Protocols" **Section:**"Enabling membership tracking and fast leave"
MLDv1/v2	MLDv2 supports source filtering, and 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.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IPv6 Multicast Features" **Section:**"MLD version distinctions"

IP service, security, and Layer 4 enhancements

TABLE 16 IP service, security, and Layer 4 enhancements

Enhancement	Description	See...
Root Guard	This is a security feature that allows a port to run STP but not allow the connected device to become the Root.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring Spanning Tree Protocol" **Section:**"STP root guard"
BPDU Guard	BPDU Guard is an extension to the port fast feature If a port is in port fast mode of operation and a BDPU is received, the port is put into the disabled mode.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring Spanning Tree Protocol" **Section:**"Spanning Tree Protocol (STP) BPDU guard"

TABLE 16 IP service, security, and Layer 4 enhancements (Continued)

Enhancement	Description	See...
Port Security MAC Violation Limit	This feature provides protection against physical link instability. It allows a user to configure it to keep a port in a down state in cases where the port has experienced some number of state transitions within a configured amount of time.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Using the MAC Port Security Feature" **Section:** "Port security MAC violation limit"
IPv6 DHCP Gateway	You can allow a DHCP client to send a message to a DHCP server by using a DHCP relay agent.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Inspecting and Tracking DHCP Packets" **Section:** "DHCP relay agent information (DHCP option 82)"

Network management

TABLE 17 Network management

Enhancement	Description	See...
IPv6 Management TFTP, SSH, Telnet, AAA, and WEB	You can perform system management tasks for the BigIron RX using the TFTP, telnet, AAA, and Secure Shell (SSH).	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring Basic IPv6 Connectivity"

Summary of enhancements and configuration notes in 02.2.01

Hardware enhancements

TABLE 18 Hardware enhancements

Enhancement	Description	See page
New Hardware Support	The following new hardware is supported with the 02.2.01 software release for the device: • Management module with 2 GB of memory • 24-port 100/1000 Mbps SFP Ethernet interface module • 48-port 1 Gbps Copper Ethernet interface module • DC Power Supply • New fan controller	**Book:** *Brocade BigIron RX Series Installation Guide*

Layer 2 enhancements

TABLE 19 Layer 2 enhancements

Enhancement	Description	See page
VLAN Byte Accounting	With this release, you can configure a VLAN to account for the number of bytes received by all the member ports.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"VLANs" **Section:**"VLAN byte accounting"
Super Aggregated VLANs (SAV)	Multiple VLANs can be aggregated within another VLAN to allow you to construct Layer 2 paths and channels.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"VLANs" **Section:**"Configuring super aggregated VLANs"
Enhancement to the **lacp system-priority** command	The **lacp system-priority** command has been moved from the interface configuration level to the global configuration level.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** See the Dynamic Link Aggregation chapter in the *BigIron RX Series Configuration Guide* - Versions 02.5.00 and earlier. **Section:**Configuring Link Aggregation Parameters

Layer 3 enhancements

TABLE 20 Layer 3 enhancements

Enhancement	Description	See page
Graceful Restart	With this release, you can enable Graceful Restart for OSPF and BGP	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring OSPF Version 2 (IPv4)" and "Configuring BGP4 (IPv4 and IPv6)" **Section:** "OSPF graceful restart" and "Graceful restart in BGP"
BGP Null0 Routing	With this release, BGP can use null0 to resolve the next hop and install null0 BGP routes to the routing table	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring BGP4 (IPv4 and IPv6)" **Section:**"BGP Null0 routing"
GRE IP Tunneling	This release supports creation of a GRE tunnel across an IP network.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP" **Section:**"GRE IP tunnel"

TABLE 20 Layer 3 enhancements (Continued)

Enhancement	Description	See page
OSPF point-to-point	OSPF point-to-point eliminates the need for Designated and Backup Designated routers, allowing for faster convergence of the network.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring OSPF Version 2 (IPv4)" **Section:** "OSPF point-to-point links"
Neighbor Local AS	Neighbor Local Autonomous System (AS) feature allows a router that is a member of one AS to appear to be a member of another AS.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring BGP4 (IPv4 and IPv6)" **Section:** "Neighbor local-AS"
Full AS Path information in sFlow	In this release, sFlow packets now contain full AP Path information.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "sFlow" **Section:** "Extended gateway information"
Policy Based Routing	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.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Policy-Based Routing"

Multicast enhancement

TABLE 21 Multicast enhancement

Enhancement	Description	See page
IGMP Snooping	The device supports IGMP snooping.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring IP Multicast Traffic Reduction" **Section:** "Enabling IP multicast traffic reduction"

Security enhancements

TABLE 22 Security enhancements

Enhancement	Description	See page
Multi-device Port Authentication	Multi-device port authentication is now supported on the device.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Using the MAC Port Security Feature"
802.1x Port Security	This release allows you to enable 802.1X port security and multi-device port authentication on the same interface.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Configuring 802.1x Port Security"

TABLE 22 Security enhancements (Continued)

Enhancement	Description	See page
Port Security MAC Deny	With this release, you can configure deny mac addresses on a global level or on a per port level.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Using the MAC Port Security Feature"
IP Fragmentation Protection	Fragmented IP packets with undersized fragments and overlapping fragments are dropped.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP" **Section:**"IP fragmentation protection"
IP Option Attack Prevention	Packets with IP options in their header are automatically dropped. Enabling the **ip ip-option-process** command allows the device to process packets that use IP options.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP" **Section:**"IP option attack protection"
IP Receive ACLs	You can use IPv4 ACLs to filter the packets intended for the management processor to protect the management module from being overloaded with heavy traffic that was sent to one of the Layer 3 Switch IP interfaces.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Access Control List" **Section:**"Specifying the destination mirror port for IP receive ACLs"
Static Route Tagging	Static routes can be configured with tag values.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring IP" **Section:**"Static route tagging"
MTU enhancements for IPv4	In this release, you can configure IPv4 MTU to be greater than 1500 bytes.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring Quality of Service" **Section:**"Changing the MTU"
Enhancements to passwords	The following have been implemented to enhance the password features in the device: • New rules for enable and user passwords • Users are now required to accept the message of the day • Users are locked out if they reach the maximum number of login attempts and have not logged in successfully. • Previous passwords used are now stored in the CLI. When users change their password, they must select a password that has not been stored in the CLI. • A password can now be set to expire	**Book:** *Brocade BigIron RX Series Installation Guide*

TABLE 22 Security enhancements (Continued)

Enhancement	Description	See page
Port Security Enhancements	You can specify how many packets from denied MAC addresses can be received on a port in a one-second interval before the device shuts the port down.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:** "Using the MAC Port Security Feature" **Section:** "Defining security violation actions"
Larger SSHv2 Crypto Key	The size of the SSH v2 crypto key in this release is larger than crypto key in previous releases. Therefore, after upgrading to this release, you must clear the existing crypto key, then regenerate a new one.	**Book:** *Brocade BigIron RX Series Installation Guide*

System enhancements

TABLE 23 System enhancements

Enhancement	Description	See page
Unified software image for software upgrades	Once the device software has been upgraded to Release 02.2.01, you can use the unified software image to upgrade the device's software.	**Book:** *Brocade BigIron RX Series Installation Guide*
Change to the SNMP MIB objects for trunking	The snMSTrunkTable has been replaced by snMSTrunkIfTable	**Book:** MIB Reference

Summary of enhancements in release 02.2.00g

TABLE 24 Summary of enhancements in 02.2.00g

Enhancement	Description	See page
New Hardware Support	The following new hardware is supported with the 02.2.01 software release for the device: • 2-port 10 Gigabit Ethernet port module • DC Power Supply	**Book:** *Brocade BigIron RX Series Installation Guide*

Summary of enhancements and configuration notes in 02.2.00

TABLE 25 Summary of emhancements in 02.2.00

Enhancement	Description	See page
Quality of Service (QoS) Support	QoS support on the device is different than for the BigIron MG8.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring Quality of Service"
Rate-limiting Support	Rate-limiting can be performed based on ACL matching of flows and L2/L3 priority. It operates as on the BigIron MG8 except: • Only Inbound rate limiting is supported. • 802.1p packet priority is used by default • Rate limit accounting is available if WRED is not enabled. CLI changes required for these differences are described in the page referenced on the next column.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring Traffic Reduction"
Hardware Forwarding of Packets	Default behavior on device is hardware unknown unicast and multicast flooding.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"VLANs" **Section:**"Hardware flooding for Layer 2 multicast and broadcast packets"
Switching and Routing Packets	Operation of packet switching and routing have changed with the device. Details are described in the page referenced on the next column.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"VLANs" **Section:**"Unknown unicast flooding on VLAN ports"
No Support for Core Device to Copy the QoS Priority	This feature is not supported on device.	N/A
Trunk Support	On the device, the switch, server, and per-packet options for trunking are not supported.	N/A
Multicast Entry Limit	1542 multicast entries are limited to IPv4 1542 entries provided every group has only one destination.	N/A
WAN PHY Mode Support	This release supports WAN PHY Mode per 10 GB Ethernet port.	**Book:** *BigIron RX Series Configuration Guide* **Chapter:**"Configuring Interface Parameters" **Section:**"Enabling WAN PHY mode support"

For further information about new features and documentation updates for this release, refer to the Knowledge Portal at *kp.foundrynet.com*.

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.

Notice to the reader

This document may contain references to the trademarks of the following corporations. These trademarks are the properties of their respective companies and corporations.

These references are made for informational purposes only.

Corporation	Referenced Trademarks and Products
HP	HP Top Tools

Related publications

The following Brocade documents supplement the information in this guide:

- *Brocade BigIron RX Series Installation Guide*.
- *MIB Reference*.

NOTE
For the latest edition of these documents, which contain the most up-to-date information, see Product Manuals at *kp.foundrynet.com*.

Getting technical help or reporting errors

Brocade is committed to ensuring that your investment in our products remains cost-effective. If you need assistance, or find errors in the manuals, contact Brocade using one of the following options:

Web access

Go to *kp.foundrynet.com* and log in to the Knowledge Portal (KP) to obtain more information about a product, or to report documentation errors. To report errors, click on Cases > Create a New Ticket. Make sure you specify the document title in the ticket description.

E-mail access

Send an e-mail to: IPsupport@brocade.com

Telephone access

United States and Canada: 800-752-8061

International: +800-ATFIBREE (+800 28 34 27 33)

Refer to the Services & Support page on www.brocade.com for additional toll-free numbers that may be available within your country.

Areas unable to access 800 numbers: +1-408-333-6061

Chapter

Getting Started with the Command Line Interface 1

In this chapter

Logging on through the CLI

NOTE
This user guide assumes that an IP address and default gateway have been assigned to the device when it was installed. If you need to assign an IP address or default gateway to the device, refer to the *Brocade BigIron RX Series Installation Guide*.

Once 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, any user who can open a direct or Telnet connection to a device Switch can access all these CLI levels. To secure access, you can configure Enable passwords or local user accounts, or you can configure the device to use a RADIUS or TACACS/TACACS+ server for authentication. Refer to the *Security Guide*.

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 followed by ?, a message appears indicating the command was unrecognized.

Example

```
BigIron RX(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 26 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.
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 device, you may see the following prompt.

```
BigIron RX>
```

The “BigIron RX“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

```
BigIron RX>enable
```

or

```
BigIron RX>enable user1 mypassword
```

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

```
BigIron RX>#.
```

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.

```
BigIron RX>enable
BigIron RX>#configuration terminal
```

The prompt changes to the Global Configuration level.

```
BigIron RX(config)#
```

CONFIG commands

CONFIG commands modify the configuration of a device. Once you are at the Global Configuration level, you can enter commands to configure the features in the device. This section describes the following 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 port-by-port basis. You reach this level by entering the following at the global CONFIG level:

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

Trunk level

The trunk level allows you to change parameters for statically-configured trunk groups. You reach this level by entering a **trunk** 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 Uuicast address family level

The global BGP and BGP4 unicast address family levels are present only on Brocade 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.

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 Global 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 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 **global-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 following prompt.

```
BigIron RX>
```

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.

NOTE
If you install switch code on a router, the command prompt begins with "SW-" to indicate the software change. This is true even if you change the system name.

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

- BigIron RX> `enable` — User Level commands
- BigIron RX# `configure terminal` — Privileged Level-EXEC commands
- BigIron RX`(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.

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

```
BigIron RX(config-if–e100-5/1)# – The interface is a 10/100 port.
BigIron RX(config-if–e1000-5/1)# – The interface is a Gigabit port.
```

NOTE
For simplicity, the port speeds sometimes are not shown in example Interface level prompts in this manual.

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.

```
BigIron RX> ? <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

To view possible **copy** command options, enter the following.

```
BigIron RX# copy ?
  flash
  running-config
  startup-config
  tftp
BigIron RX# 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 12 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.

```
BigIron RX# 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 Brocade device.

```
BigIron RX# 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 device.

```
BigIron RX# 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

```
BigIron RX# ?
  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 Brocade 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 "Using special characters in regular expressions" on page 12 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 27 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+

TABLE 27 Special characters for regular expressions (Continued)

Character	Operation
?	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.
[ ]	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. See the example above.
|	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 “*”.

```
BigIron RX# 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:

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

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.

1 Searching and filtering output

Chapter

2

Getting Familiar With the BigIron RX Series Switch Management Applications

In this chapter

How to manage BigIron RX Series switch

This chapter describes the different applications you can use to manage the BigIron RX Series Switch. The BigIron RX Series Switch supports the same management applications as other Brocade devices.

As with other Brocade devices, you can manage a BigIron RX Series Switch 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.
- **IronView Network Manager** – An optional SNMP-based standalone GUI application.

The following section describes how to log on to these applications.

Logging on through the CLI

Once an IP address is assigned to the BigIron RX Series Switch'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, any user who can open a direct or Telnet connection to a BigIron RX Series Switch can access all these CLI levels. To secure access, you can configure Enable passwords or local user accounts, or you can configure the device to use a RADIUS or TACACS/TACACS+ server for authentication. Refer to the *Security Guide*.

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 followed by **?**, a message appears indicating the command was unrecognized.

Example

```
BigIron RX(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 28 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.
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.

For a complete list of CLI commands and syntax information for each command, refer to the *Switch and Router Command Line Interface Reference*.

Searching and filtering output from CLI commands

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.

You can also 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 22 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 3/1 so it displays only lines containing the word "Internet". This command can be used to display the IP address of the interface.

```
BigIron RX# show interface e 3/1 | 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
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 a BigIron RX Series Switch.

```
BigIron RX# 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 BigIron RX Series Switch.

```
BigIron RX# 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 displays 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 to cancel the display. In addition, you can search and filter output from this prompt.

Example

```
BigIron RX# ?
  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. For 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. See 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 29 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.

TABLE 29 Special characters for regular expressions (Continued)

Character	Operation
^	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 matchoccurs if any of the characters within the range is present. See the example above.
|	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 “*”.

```
BigIron RX# 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:

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

Logging on through the Web Management Interface

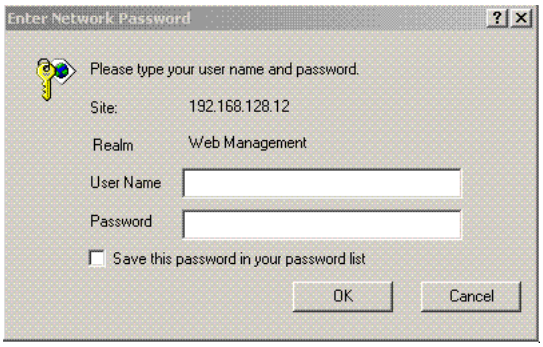
To use the Web Management Interface, open a Web browser and enter the IP address of a BigIron RX Series Switch's management port in the Location or Address field. The Web browser contacts the device and displays the login panel for the BigIron RX Series Switch, as shown in Figure 1.

FIGURE 1 Web Management Interface login panel

NOTE
If you are unable to connect with the device through a Web browser due to a proxy problem, it may be necessary to set your Web browser to direct Internet access instead of using a proxy. For information on how to change a proxy setting, refer to the on-line help provided with your Web browser.

To log in, click on the Login link. Figure 2 shows the dialog box that displays.

FIGURE 2 Web Management Interface login dialog box



The login username and password you enter depends on whether your device is configured with AAA authentication for SNMP. If AAA authentication for SNMP is not configured, you can use the user name “get” and the default read-only password “public” for read-only access. However, for read-write access, you must enter “set” for the user name, and enter a read-write community string you have configured on the device for the password. There is no default read-write community string. You must add one using the CLI. Refer to the *Security Guide*.

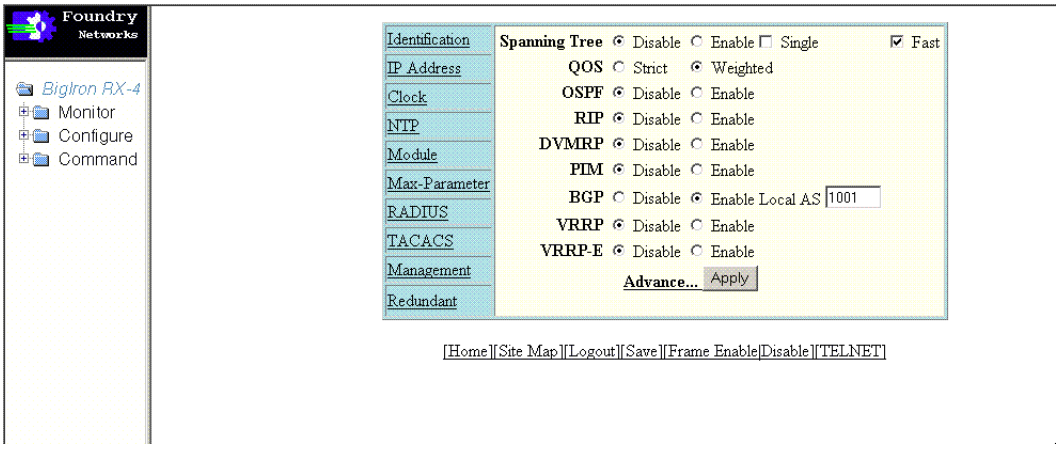
Web Management Interface

When you log into a device, the System configuration panel is displayed. This panel allows you to enable or disable major system features. You can return to this panel from any other panel by selecting the Home link.

The Site Map link gives you a view of all available options on a single screen.

Figure 3 displays the Web Management Interface panel for Layer 3 Switch features. This panel allows you to configure the features supported by the Layer 3 Switch software.

FIGURE 3 Panel for Layer 3 Switch features



The left pane of the Web Management Interface window contains a “tree view,” similar to the one found in Windows Explorer. Configuration options are grouped into folders in the tree view. These folders, when expanded, reveal additional options. To expand a folder, click on the plus sign to the left of the folder icon.

Logging on through IronView Network Manager

Refer to the *IronView Network Management User's Guide* for information about using IronView Network Manager.

Chapter

Using a Redundant Management Module 3

In this chapter

How management module redundancy works

You can install a redundant management module in slot M1 or M2 of the BigIron RX Series chassis. By default, the system considers the module installed in slot M1 to be the active management module and the module installed 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.

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 and how the switchover occurs as a result of each event.
- Implications that you should be aware of if a switchover occurs.

Management module redundancy overview

When you power on or reload a BigIron RX Series chassis with two management modules installed, by default, the management module installed in slot M1 becomes the active module and the module installed in slot M2 becomes the standby module. (You can change the default active slot from M1 to M2 using the **active-management** command. For information about performing this task, refer to "Changing the default active Chassis slot" on page 31.)

After the active and standby modules are determined, both modules boot from the source specified for the active module. The active management module can boot from the following sources:

- The active management module's flash memory.
- A PCMCIA flash card inserted in one of the PCMCIA slots in the active management module's front panel.

After the modules boot, the active module compares the standby module's flash code and system-config file to its own. If differences exist, the active module synchronizes the standby module's flash code and system-config file with its own.

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. Keeping the system-config and running-config files on both modules synchronized allows the standby module to assume the role of active module seamlessly if necessary.

The interface modules are not reset, as they are with the previous cold-restart redundancy feature. The interface modules 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 can be overwritten in some cases which can cause an interruption of traffic forwarding.

Management module switchover

The events cause the standby management module to become the active module, which is called a ***switchover***. Those events are as follows:

- 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 role of active management module from the currently active module to the standby module. For example, if the module in slot M2 is the active module and the module in slot M1 is the standby module and you want the module in M1 to be the active module and the module in M2 to be 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 34.

When the switchover occurs, the standby module becomes the active module.

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 role of the active module. After you insert a replacement module in the slot from which the original active module was removed, the replacement module becomes the standby module. The module boots from the source specified for the active module. The active management module can boot from the following sources:

- The active management module's flash memory.
- A PCMCIA flash card inserted in one of the PCMCIA slots in the active management module's front panel.

After the replacement module boots, the active module compares the standby module's flash code and system-config file to its own. If differences exist, the active module synchronizes the standby module's flash code and system-config file with its own.

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 as is. Communication between the active module and the removed module stops until the new module is installed in the BigIron RX Series chassis. After the new module is installed, it assumes the role of standby module. The module boots from the source specified for the active module. The active management modules can boot from the following sources:

- The active management module's flash memory.
- A PCMCIA flash card inserted in one of the PCMCIA slots in the active management module's front panel.

After the module boots, the active module compares the standby module's flash code and system-config file to its own. If differences exist, the active module synchronizes the standby module's flash code and system-config file with its own.

Switchover implications

After the role of the active management module switches from one module to another, you must be aware of implications that affect the following areas:

- Management sessions
- Syslog and SNMP traps
- MAC addresses

The following sections explain the implications for these areas.

Management sessions

You can establish management sessions with the active management module's management port. If a switchover occurs, the management port on the original active module shuts down and all open CLI, Web management interface, and *IronView Network Manager* sessions with that port close. You can open new sessions with the new active module, provided that the new active 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 module's management port, you can open a new session if a PC is connected to the new active module's management port.)

In the scenario described above, you can open a new session using the same IP address you were using before the switchover. (You configure an IP address for the active module only; if a switchover occurs, the IP address is used by the new active module.)

Syslog and SNMP traps

When a switchover occurs, the BigIron RX system sends a Syslog message to the local Syslog buffer and also to the Syslog server, if you have configured the system to use one. In addition, if you have configured an SNMP trap receiver, the system sends an SNMP trap to the receiver.

When the system is powered on or otherwise reset normally, the system sends a cold start message and trap. However, if the system is reset as the result of switchover to the standby management module, the system instead sends a warm start message and trap.

MAC address changes

The MAC addresses in theBigIron RX Series system are based on the MAC address of the BigIron RX Series chassis. During switchover, the system's MAC addresses change and the system sends out gratuitous ARP requests to flush the old MAC addresses from the ARP caches on attached IP devices, and update the caches with the system's new MAC addresses.

Layer 2 Hitless Failover

The Layer 2 Hitless Failover feature provides automatic failover from the active management module to the standby management module without interrupting operation of any interface modules in the chassis. 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.

NOTE
Since both the standby and active management modules run the same code, a command that brings down the active management module will most likely bring down the standby management module. Because all configuration commands are synchronized from active to standby management module in real time, both management modules will crash at almost the same time. This in turn causes the system to reset all interface modules (similar to the behavior when the 'reboot' command is executed) and causes packet loss associated with a system reboot.

Once booted, the redundant management module keeps up-to-date copies of the active module's running configuration. Layer 2 protocols such as STP, RSTP, MRP, and VSRP are run concurrently on both the active and standby management modules. Upon the failover of the active management module, the standby module takes over as the active management module and picks up where the active module left off, without interrupting any Layer 2 traffic.

The interface modules are not reset, as they are with the previous cold-restart redundancy feature. The interface modules 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 can be overwritten in some cases which can cause an interruption of traffic forwarding. Layer 3 hitless failover is not supported in this release. Consequently, a failover will result in a re-synchronization of Layer 3 data structures

Management module redundancy configuration

Configuring management module redundancy consists of performing one optional task (changing the default active chassis slot). The section explains how to perform this task.

Changing the default active Chassis slot

By default, the BigIron RX Series system considers the module installed in slot M1 to be the active management module. If desired, you can change the default active chassis slot to M2.

The **active-management** command determines which management module will become active after a power cycle. By default, the top or left mgmt module will become active after power cycle. This information is stored in the chassis's backplane eeprom, and not in the configuration file. This is a chassis specific configuration.

To change the default active chassis slot to M2, enter the following commands.

```
BigIron RX(config)# redundancy
BigIron RX(config-redundancy)# active-management mgmt-2
```

Syntax: active-management <mgt-module>

The <mgt-module> parameter specifies the management module, which can be 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 MPs are in the chassis.

Managing management module redundancy

The BigIron RX Series Switch allows you to perform the following management tasks related to management module redundancy:

- Perform immediate synchronization of files.
- Perform a manual switchover to the standby module.
- Reboot the standby module.

File synchronization between the active and standby management modules

Each active and standby management module contains the following files that can be synchronized between the two modules are:

- **Flash code** – The flash code can include the following files:
  - monitor, which contains the management module's Real Time Operating System (RTOS).
  - primary, which contains the management module's primary RX Series IronWare image.
  - secondary, which contains the management module's secondary RX Series IronWare image.

  A RX Series IronWare image contains the layer 1 – 3 software run by the management module.

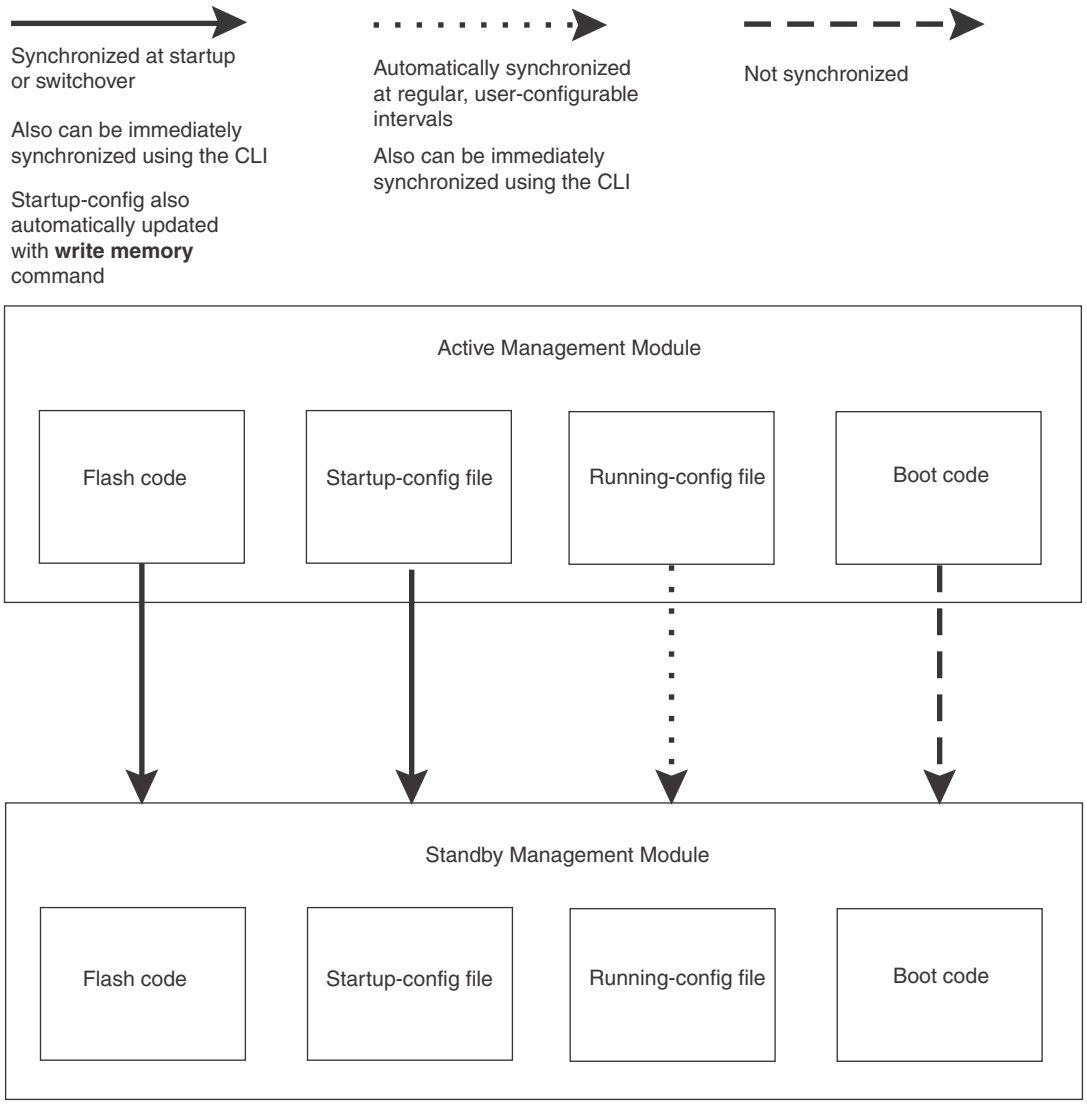
  During startup or switchover, the active module compares the standby module's flash code to its own. If differences exist, the active module synchronizes the standby module's flash code with its own. If you update the flash code on the active module, the active module automatically synchronizes (without comparison) the standby module's flash code with its own.
- **System-config file** – The flash code also includes the system-config file. During startup or switchover, the active module compares the standby module's system-config file to its own. If differences exist, the active module synchronizes the standby module's system-config file with its own. When you save changes to the system-config file on the active module, the active module automatically synchronizes (without comparison) the standby module's system-config file with its own.
- **Running-config** – The running-config file resides in the BigIron RX Series system's memory. The running-config file is automatically synchronized (without comparison) from the active module to the standby module at regular intervals. The default interval is 7 seconds.

Each active and standby management module also includes boot code, which is the code a module runs when it first starts up. The boot code resides in each module's boot flash. The boot code is not synchronized between the two modules. The unsynchronized boot code allows the system to run using an older version of boot code on the standby module if desired.

NOTE
Whenever you load a new card into a chassis, check the card to ensure that it has the appropriate boot code revision. If the boot code is from a previous version, then upgrade the code.

Figure 4 shows how the files are synchronized between the active module and the standby module.

FIGURE 4 Active and standby management module file synchronization



The BigIron RX Series system allows you to do the following related to file synchronization:

- 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 file, and running-config file on the active management module with the same files on the standby module and synchronize the files immediately if differences exist. When you synchronize the files, the active module copies its files to the standby module, replacing the files on the standby module.

To compare and immediately synchronize files between the active and standby modules if differences exist, enter the following command at the Privileged EXEC level of the CLI.

```
BigIron RX# sync-standby
```

Syntax: 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, the active module copies its files 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 of the CLI.

```
BigIron RX# force-sync-standby
```

Syntax: force-sync-standby

Manually switching over to the standby management module

You can cause the BigIron RX Series system to switch over to the standby module (and thus make it the active module). To do so, you can enter either the **switchover** or the **reset** commands at the Privileged EXEC level.

```
BigIron RX# switchover
```

or

```
BigIron RX# reset
```

Syntax: switchover

Syntax: reset

Rebooting the active and standby management modules

You can reboot the management modules, maintaining the active and standby roles currently performed by each module, using the **boot system** or **reload** commands. You can also reboot the standby module only, maintaining its current standby role, using the **reboot-standby** command.

For example, to reboot the active and standby management modules from the primary RX Series IronWare image in the management module's flash memory, enter the following command at the Privileged EXEC level.

```
BigIron RX# boot system flash primary
```

Syntax: boot system bootp | [flash primary | flash secondary] | slot <number> <filename> | tftp <ip-address> <filename>

The **flash primary** keyword specifies the primary RX Series IronWare image in the management module's flash memory, while the **flash secondary** keyword specifies the secondary RX Series 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 BigIron RX Series Switch to boot from an RX Series 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.

```
BigIron RX# reload
```

Syntax: reload

To reboot the standby module only, enter the following command at the Privileged EXEC level.

```
BigIron RX# reboot-standby
```

Syntax: reboot-standby

Monitoring management module redundancy

You can monitor the following aspects of management module redundancy:

- The status of the management modules (if a module is the active or standby module).
- The switchover history for the management modules.

The following sections explain how you can monitor the management modules.

Determining management module status

You can determine the status of a management module in the following ways:

- **LEDs** – The management module's LEDs indicate whether a module is the active module or the standby module, and if the module has power.
- **Module information in software** – The module information displayed by the software indicates whether a module is the active module or the standby module.

Status LED

If you are located near the BigIron RX Series chassis, you can determine which management module is currently the active module and which is the standby module 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 this LED is off, the module is not receiving power. (A module without power will not function as the active or standby module.)

Software

To display the status of the management modules, enter the following command at any CLI level.

```
BigIron RX# show module
       Module                                    Status     Ports  Starting MAC
M1 (upper): BigIron BI-RX Management Module  Active
M2 (lower): BigIron BI-RX Management Module  Standby (Ready)
...
```

Syntax: show module

The Status column indicates the module status. The management modules can have one of the following status:

- **ACTIVE** – The module is currently the active management module.
- **STANDBY** – The module is the standby management module. The status of the standby module can be one of the following:
  - **Init** – The module is currently initializing as the standby module.
  - **Ready** – The module is ready to take over as the active module, if necessary.
  - **Wait** – The module is awaiting boot information from the active management module.
  - **Sync** – The active module is currently synchronizing files between itself and the standby module.

Displaying temperature information

Each management module contains a temperature sensor. By default, the BigIron RX system polls the temperature of each management module every 60 seconds. You can display the current temperature of the management modules (and all other modules) by entering the following command at any CLI level.

```
BigIron RX# show chassis
...
Active Mgmt Module: 28.43C 57.500C (CPU)
Standby Mgmt Module: 29.15C  57.500C (CPU)...
Temperature Monitoring Poll Period is 60 seconds
...
```

Syntax: show chassis

The output displays the temperature of the management modules in the BigIron RX chassis and also indicates that the temperature readings were provided within the last 60 seconds.

Displaying switchover information

You can display the following related to a switchover:

- Redundancy parameter settings and statistics, which include the number of switchover that have occurred.
- System log or the traps logged on an SNMP trap receiver, which includes 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.

```
BigIron RX# show redundancy
=== MP Redundancy Settings ===
Default Active Slot = 17
Running-Config Sync Period = 7 seconds

=== MP Redundancy Statistics ===
Current Active Session:
Active Slot = 9,Standby Slot = 10 (Ready State),Switchover Cause = No Switchover
Start Time = 0-0-17  19:47:39 (Wednesday)

Previous Active Session #1:
Active Slot = 10, Standby Slot = 9, Switchover Cause = Active Rebooted
Start Time = 0-0-17  19:46:9 (Wednesday), End Time = 0-0-17  19:47:39 (Wednesday)

Previous Active Session #2:
Active Slot = 9, Standby Slot = 10, Switchover Cause = Active Rebooted
Start Time = 0-0-17  19:44:14 (Wednesday), End Time = 0-0-17  19:46:9 (Wednesday)
...
```

This output displays that the default active chassis slot is configured as slot 9 (M1) and the automatic synchronization interval is configured for 7 seconds. It also displays that in the current active session, the module installed in slot 9 (M1) is the active module, the module installed in slot 10 (M2) is the standby module, which is in Ready state, and no switchovers have occurred.

However, in two previous sessions, switchovers occurred because the active module was rebooted. In session #2, the module installed in slot 9 (M1) was the active module, while the module installed in slot 10 (M2) was the standby module. In session #1, the module installed in slot 10 (M2) was the active module, while the module installed in slot 9 (M1) was the standby module.

To view the system log or the traps logged on an SNMP trap receiver, enter the following command at any level of CLI.

```
BigIron RX# 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 displays that one switchover occurred.

Flash memory and PCMCIA flash card file management commands

The BigIron RX Series system supports file systems in the following locations:

- The management module's flash memory.
- A PCMCIA flash card inserted in the management module's slots 1 or 2.

Table 30 outlines the root directory for each file system.

TABLE 30 BigIron RX 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. You can 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 the 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 or "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.

In the CLI, you can access all the 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 flash card's slot 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 the card erases all data stored on the card.

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 BigIron RX Series system, by default, the management focus is on flash memory.

You can change the current management focus from flash memory to a slot and subdirectory using the **cd** or **chdir** command. (For more information about these commands, refer to "Switching the management focus" on page 43.)

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 42.)

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** or **mkdir** and **rd** or **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 be 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** or **mkdir** and **rd** or **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 a file's location 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.

```
BigIron RX# 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 . 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>). When using 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 are not shipped with a management module If you want to use a flash card, you must formatted it for the 16 FAT file system before you can store files on the card.

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.

For example, to reformat a flash card in the management module's slot 2, enter the following command.

```
BigIron RX# 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 39.

If you are not sure which file system has the current management focus, enter the following command.

```
BigIron RX# pwd
Flash /flash/
```

Syntax: pwd

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.

```
BigIron RX# pwd
/slot1/
```

In the following example, the management focus is a subdirectory called "test" on the flash card in slot 1.

```
BigIron RX# 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 management module's flash memory. You can switch the focus from flash memory to the management module's slot 1 or slot 2 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.

```
BigIron RX# cd /slot2
BigIron RX#
```

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.

```
BigIron RX# cd /slot1
Device not present
```

Syntax: cd <directory-pathname>

Syntax: chdir <directory-pathname>

For the <directory-pathname> parameter for both **cd** and **chdir** commands, you can 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 a 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.

```
BigIron RX# cd /PLOOK
```

If you specify an invalid subdirectory path, the CLI displays a message such as the following.

```
BigIron RX# cd /PLOOK
Path not found
```

If you are certain the path you specified exists, make sure you are at the correct level for reaching 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.

```
BigIron RX# cd /flash
BigIron RX#
```

Displaying a directory of the files

You can display a directory of the files in the management module's flash memory or on a flash card inserted in the management module's 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.

```
BigIron RX# 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 or 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.

```
BigIron RX# 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.

```
BigIron RX# 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 31 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 Brocade device's 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 Brocade device's system clock is set.
File size	The number of bytes in the file.
Read-write attribute	If you have set the file's read-write attribute to read-only, "R" appears before the file name. If the file's read-write attribute 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 49.
File name	The file name.
Long file name	This field applies to files on a flash card only. The longer file name 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 management module's flash memory or on a flash card inserted in the management module's slot 1 or slot 2.

The software attempts to display the specified 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 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.

```
BigIron RX# 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.

```
BigIron RX# more /slot2/cfg.cfg
```

Displaying the hexadecimal output of a file

You can display the hexadecimal output of a file in the management module's flash memory or on a flash card inserted in the management module's slot 1 or slot 2.

The software attempts to display the hexadecimal output of a specified 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 display the hexadecimal output of 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 **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.

```
BigIron RX# hd cfg.cfg
```

Syntax: 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.

```
BigIron RX# hd /slot2/cfg.cfg
```

Creating a subdirectory

You can 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 attempts to create a subdirectory 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 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.

```
BigIron RX# mkdir slot2 TEST
```

Syntax: 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. 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.

```
BigIron RX# 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 43.

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 attempts to remove a subdirectory from 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 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.

```
BigIron RX# rmdir slot2 TEST
```

Syntax: 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.

```
BigIron RX# 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 42.

Renaming a file

You can rename a file in the management module's flash memory or on a flash card inserted in the management module's slot 1 or slot 2 using the **rename** or **mv** command.

The software attempts to rename 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 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.

```
BigIron RX# rename oldname newname
```

If the command is successful, the CLI displays a new command prompt.

Syntax: 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 such as the following.

```
BigIron RX# 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
The read-write attribute of all files in flash memory is set to read-write. You cannot change this attribute for the files in flash memory. 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 43.

The software attempts to change the read-write attribute of 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 change this attribute of the file 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 such as the following.

```
BigIron RX# attrib slot2 ro goodcfg.cfg
```

Syntax: 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 files 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 such as the following.

```
BigIron RX# 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 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. When deleting a file from flash memory, make sure you really want to delete the file.

The software attempts to 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 such as the following.

```
BigIron RX# 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 files 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 such as the following.

```
BigIron RX# 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.

```
BigIron RX# delete /slot2/
```

or

```
BigIron RX# delete slot2
```

Recovering (“undeleting”) a file

You can recover or undelete a file you have deleted from a flash card file system.

NOTE
You cannot 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 attempts to recover the file in the file system that has the current management focus. By default, flash memory has the management focus. 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 43.

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.

```
BigIron RX# cd slot2
BigIron RX# 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 the CTRL + C key combination.

Syntax: undelete

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 attempts to append one file to another 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 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 such as the following.

```
BigIron RX# append newacls.cfg startup-config.cfg
```

Syntax: 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 such as the following.

```
BigIron RX# 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, see Basic Tasks in the Software Upgrade Process in the *Brocade BigIron RX Series 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 management module's flash memory.
- Copy software images between active and standby management modules.
- Copy files from a management module to an interface module.
- Copy management module RX Series 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 the management module's flash memory.
- Copy a startup-config file between the management module's flash memory 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 device's running-config (loading ACLs only)

NOTE
The copy options require you to explicitly specify the flash card. Therefore, 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.

```
BigIron RX# 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 such as the following.

```
BigIron RX# 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 such as the following.

```
BigIron RX# 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 RX Series 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.

```
BigIron RX# copy flash flash monitor standby
```

Syntax: copy flash flash monitor standby

To copy the RX Series IronWare image from the secondary location in the active management module's flash memory to the primary location in the active module's flash memory, enter the following command.

```
BigIron RX# copy flash flash primary
```

Syntax: copy flash flash primary [standby]

Specify the optional **standby** keyword to copy the RX Series IronWare image from the secondary location in the active management module's flash memory to the primary location in the standby module's flash memory.

To copy the RX Series IronWare image from the primary location in the active management module's flash memory to the secondary location in the active module's flash memory, enter the following command.

```
BigIron RX# copy flash flash secondary
```

Syntax: copy flash flash secondary [standby]

Specify the optional **standby** keyword to copy the RX Series IronWare image from the primary location in the active management module's flash memory to the secondary location in the standby module's flash memory.

Copying files from a management module to an interface module

You can copy a software image or other type of file from the management module's flash memory to the flash memory of one or all interface modules.

For example, to copy the interface module's monitor image from the management module to all interface modules, enter a command such as the following.

```
BigIron RX# 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 such as the following.

```
BigIron RX# copy flash lp test.cfg lptest.cfg 1
```

Syntax: copy flash lp <source-file> <dest-file> <slot-number> | all

Copying RX Series IronWare images from flash memory to a TFTP server

You can copy the management module's RX Series IronWare images from the primary and secondary locations in flash memory to a TFTP server.

For example, to copy the RX Series IronWare image in the secondary location in flash memory to a TFTP server, enter a command such as the following.

```
BigIron RX# 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

You can use the following methods to copy files between a flash card and a TFTP server.

NOTE
The BigIron RX Series system must have network access to the TFTP server.

To copy a file from a flash card to a TFTP server, enter a command such as the following.

```
BigIron RX# 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 such as the following.

```
BigIron RX# 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 RX Series 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 BigIron RX Series Switch uses the startup-config in the primary area of flash memory to configure itself when you boot or reload the device.

NOTE
The BigIron RX Series Switch cannot use a startup-config file on a flash card to configure itself. 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 such as the following.

```
BigIron RX# 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 device's flash memory. The next time you reboot or reload the device, it uses the configuration information in test2.cfg.

To copy the device's startup-config file from flash memory onto a flash card, enter a command such as the following.

```
BigIron RX# 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 device's flash memory 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 BigIron RX Seriessystem has access. By default, the device uses the startup-config in the primary area of flash memory to configure itself when you boot or reload the device.

To copy the device's startup-config from flash memory to a TFTP server, enter a command such as the following.

```
BigIron RX# 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 such as the following.

```
BigIron RX# 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 BigIron RX Series Switch's running-config to a flash card or a TFTP server. The running-config contains the device's currently active configuration information. When you copy the running-config to a flash card or TFTP server, you are making a copy of the device's current configuration, including any configuration changes you have not saved to the startup-config.

To copy the device's running configuration into a file on a flash card, enter a command such as the following.

```
BigIron RX# copy running-config slot1 runip.1
```

Syntax: copy running-config slot1 | slot2 [/<to-dir-path>/]<to-name>

To copy the device's running configuration into a file on a TFTP server, enter a command such as the following.

```
BigIron RX# 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 BigIron RX Series Switch's active configuration.

NOTE
A configuration file that you create must follow the same syntax rules as the startup-config the device creates. Refer to the "Upgrading Software Images and Configuration Files" chapter in the *Brocade BigIron RX Series Installation Guide* for additional information.

To copy a running-config from a flash card, enter a command such as the following.

```
BigIron RX# copy slot2 running-config runacl.2
```

Syntax: copy slot1 | slot2 running-config [/<from-dir-path>/]<from-name>

The command in this example changes the device's active configuration based on the information in the file.

To copy a running-config from a TFTP server, enter a command such as the following.

```
BigIron RX# 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 device's active configuration.

Copying files using the cp command

Using the **cp** command, you can 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 attempts to copy a file in a file system to another location 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 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 such as the following.

```
BigIron RX# 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 such as the following.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# cp /slot1/cfg/new.cfg /slot2/cfg/new.cfg
```

Loading the software

By default, the management module loads its RX Series IronWare image from the primary location in flash memory. You can change the system's RX Series IronWare image source to one of the following sources for one 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 RX Series IronWare image is unavailable, the system uses the primary location in flash memory as a default backup source.

Rebooting from the system

To use another source instead of the RX Series IronWare image in the primary location in flash memory for one reboot, enter a command such as the following at the Privileged EXEC level of the CLI.

```
BigIron RX# boot system slot1 /slot1/nmpr02200.bin
```

The command in this example reboots the system using the image nmpr02200.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 nmpr02200.bin on a TFTP server, enter a command such as the following.

```
BigIron RX# boot system tftp 10.10.10.1 nmpr02200.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 RX Series IronWare image on the TFTP server.

For example, to reboot the system using the secondary location in flash memory, enter the following command.

```
BigIron RX# boot system flash secondary
```

Syntax: boot system flash secondary

To reboot the system from a BOOTP server, enter the following command.

```
BigIron RX# boot system bootp
```

Syntax: boot system bootp

Configuring the boot source for future reboots

To change the RX Series IronWare image source from the primary location in flash memory to another source for future reboots, enter a command such as the following at the global CONFIG level of the CLI.

```
BigIron RX(config)# boot system slot1 nmpr02200.bin
```

The command in this example sets PCMCIA slot 1 as the primary boot source for the BigIron RX Switch. When you reload the software or power cycle the device, the device will look for the RX Series 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.

The device's response 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 BigIron RX Series 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.

```
BigIron RX# locate startup-config
Startup-config data location is flash memory
```

Syntax: locate startup-config

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. If you want to change the save location to a flash card in slot 1 or 2, enter a command such as the following.

```
BigIron RX# locate startup-config slot1 switch1.cfg
BigIron RX# 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 switch1.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 switch1.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 **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 such as the following.

```
BigIron RX# locate startup-config flash-memory switch1.cfg
BigIron RX# write memory
```

File management messages

The following table lists the messages the CLI can display in response to file management commands.

TABLE 32 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 the command 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 if you removed 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 can show up 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

Chapter

4

Securing Access to Management Functions

In this chapter

Securing access methods

This chapter explains how to secure access to management functions on the device.

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

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

TABLE 33 Ways to secure management access to the device

Access method	How the access method is secured by default	Ways to secure the access method	See page
Serial access to the CLI	Not secured	Establish passwords for management privilege levels	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 71
		Establish passwords for management privilege levels	page 72
		Set up local user accounts	page 75
		Configure TACACS/TACACS+ security	page 80
		Configure RADIUS security	page 96

TABLE 33 Ways to secure management access to the device (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
Telnet access	Not secured	Regulate Telnet access using ACLs	page 64
		Allow Telnet access only from specific IP addresses	page 67
		Allow Telnet access only to clients connected to a specific VLAN	page 68
		Specify the maximum number of login attempts for Telnet access	page 68
		Disable Telnet access	page 70
		Establish a password for Telnet access	page 71
		Establish passwords for privilege levels of the CLI	page 72
		Set up local user accounts	page 75
		Configure TACACS/TACACS+ security	page 80
		Configure RADIUS security	page 96
Secure Shell (SSH) access	Not configured	Configure SSH	page 867
		Regulate SSH access using ACLs	page 64
		Allow SSH access only from specific IP addresses	page 67
		Establish passwords for privilege levels of the CLI	page 72
		Set up local user accounts	page 75
		Configure TACACS/TACACS+ security	page 80
		Configure RADIUS security	page 96
Web management access	SNMP read or read-write community strings	Regulate Web management access using ACLs	page 65
		Allow Web management access only from specific IP addresses	page 67
		Allow Web management access only to clients connected to a specific VLAN	page 69
		Disable Web management access	page 70
		Configure SSL security for the Web management interface	page 78
		Set up local user accounts	page 75
		Establish SNMP read or read-write community strings for SNMP versions 1 and 2	page 1001
		Establishing user groups for SNMP version 3	page 1006
		Configure TACACS/TACACS+ security	page 80
		Configure RADIUS security	page 96

TABLE 33 Ways to secure management access to the device (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
SNMP (*IronView Network Manager*) 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.	Regulate SNMP access using ACLs	page 65
		Allow SNMP access only from specific IP addresses	page 67
		Disable SNMP access	page 70
		Allow SNMP access only to clients connected to a specific VLAN	page 69
		Establish passwords to management levels of the CLI	page 72
		Set up local user accounts	page 75
		Establish SNMP read or read-write community strings	page 80
TFTP access	Not secured	Allow TFTP access only to clients connected to a specific VLAN	page 69

Restricting remote access to management functions

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

- Using ACLs to restrict Telnet, Web management interface, 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, Web management interface, or SNMP access to the device

Using ACLs to restrict remote access

You can use standard ACLs to control the following access methods to management functions on the device:

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

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
2. Configure a Telnet access group, SSH access group, web access group, and SNMP community strings. 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. See the asdf chapter 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.

```
BigIron RX(config)# access-list 10 deny host 209.157.22.32 log
BigIron RX(config)# access-list 10 deny 209.157.23.0 0.0.0.255 log
BigIron RX(config)# access-list 10 deny 209.157.24.0 0.0.0.255 log
BigIron RX(config)# access-list 10 deny 209.157.25.0/24 log
BigIron RX(config)# access-list 10 permit any
BigIron RX(config)# telnet access-group 10
BigIron RX(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: telnet access-group <num> | <name>

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

The <name> parameter specifies the standard access list name.

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

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

The ACL in the example permits Telnet access only to the IP 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.

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

Syntax: ssh access-group <num> | <name>

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

The <name> parameter specifies the standard 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 IP 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 an ACL to restrict Web management access

To configure an ACL that restricts Web management access to the device, enter commands such as the following.

```
BigIron RX(config)# access-list 12 deny host 209.157.22.98 log
BigIron RX(config)# access-list 12 deny 209.157.23.0 0.0.0.255 log
BigIron RX(config)# access-list 12 deny 209.157.24.0/24 log
BigIron RX(config)# access-list 12 permit any
BigIron RX(config)# web access-group 12
BigIron RX(config)# write memory
```

Syntax: web access-group <num> | <name>

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

The <name> parameter specifies the standard access list name.

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

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.

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

The 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: snmp-server community <string> ro | rw
<standard-acl-name> | <standard-acl-id>

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

NOTE
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 <standard-acl-name> | <standard-acl-id> parameter specifies which ACL will be used to filter incoming SNMP packets.

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

The <standard-acl-name> parameter specifies the standard 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.

Configuring hardware-based remote access filtering on the device

The following is an example of configuring the device to perform hardware filtering for Telnet access.

```
BigIron RX(config)# vlan 3 by port
BigIron RX(config-vlan-3)# untagged ethe 3/1 to 3/5
BigIron RX(config-vlan-3)# router-interface ve 3
BigIron RX(config-vlan-3)# exit
BigIron RX(config)# interface ve 3
BigIron RX(config-ve-1)# ip address 10.10.11.1 255.255.255.0
BigIron RX(config-ve-1)# exit
BigIron RX(config)# access-list 10 permit host 10.10.11.254
BigIron RX(config)# access-list 10 permit host 192.168.2.254
BigIron RX(config)# access-list 10 permit host 192.168.12.254
BigIron RX(config)# access-list 10 permit host 192.64.22.254
BigIron RX(config)# access-list 10 deny any
BigIron RX(config)# telnet access-group 10 vlan 3
BigIron RX(config)# ssh access-group 10 vlan 3
BigIron RX(config)# web access-group 10 vlan 3
BigIron RX(config)# snmp-server community private rw 10 vlan 3
```

In this example, a Layer 3 VLAN is configured as a remote-access management VLAN and a router interface. The IP address specified for the router interface becomes the management IP address of the VLAN.

Restricting remote access to the device to specific IP addresses

By default, a device 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
- Web Management 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 device only to the host with IP address 209.157.22.39, enter the following command.

```
BigIron RX(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 device only to the host with IP address 209.157.22.39, enter the following command.

```
BigIron RX(config)# ip ssh client 209.157.22.39
```

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

Restricting Web Management access to a specific IP address

To allow Web Management access to the device only to the host with IP address 209.157.22.26, enter the following command.

```
BigIron RX(config)# web client 209.157.22.26
```

Syntax: [no] web client <ip-addr>

Restricting SNMP access to a specific IP address

To allow SNMP access (which includes *IronView Network Manager*) to the device only to the host with IP address 209.157.22.14, enter the following command.

```
BigIron RX(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, Web, and SNMP management access to the device 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.

```
BigIron RX(config)# all-client 209.157.22.69
```

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

Specifying the maximum number of login attempts for Telnet access

If you are connecting to the device using Telnet, the device prompts you for a username and password. By default, you have up to3 chances to enter a correct username and password. If you do not enter a correct username or password after 3 attempts, the device disconnects the Telnet session.

You can specify the number of attempts a Telnet user has to enter a correct username and password before the device disconnects the Telnet session. For example, to allow a Telnet user up to 3 chances to enter a correct username and password, enter the following command.

BigIron RX(config)# telnet login-retries 5

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

You can specify from 0 – 3 attempts. The default is 3 attempts.

Restricting remote access to the device to specific VLAN IDs

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

- Telnet access
- Web management 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.

```
BigIron RX(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 Web management access to a specific VLAN

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

```
BigIron RX(config)# web-management enable vlan 10
```

The command configures the device to allow Web 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] web-management 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.

```
BigIron RX(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.

```
BigIron RX(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>

Disabling specific access methods

You can specifically disable the following access methods:

- Telnet access
- Web Management access
- SNMP access

NOTE
If you disable Telnet access, you will not be able to access the CLI except through a serial connection to the management module, nor will you be able to use some of the features in *IronView Network Manager*. If you disable SNMP access, you will not be able to use *IronView Network Manager* or third-party SNMP management applications.

Disabling Telnet access

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

```
BigIron RX(config)# no telnet-server
```

To re-enable Telnet operation, enter the following command.

```
BigIron RX(config)# telnet-server
```

Syntax: [no] telnet-server

Disabling Web management access

If you want to prevent access to the device through the Web management interface, you can disable the Web management interface.

NOTE
As soon as you make this change, the device stops responding to Web management sessions. If you make this change using your Web browser, your browser can contact the device, but the device will not reply once the change takes place.

To disable the Web management interface, enter the following command.

```
BigIron RX(config)# no web-management
```

To re-enable the Web management interface, enter the following command.

```
BigIron RX(config)# web-management
```

Syntax: [no] web-management

Disabling Web management access by HP ProCurve Manager

By default, TCP ports 80 is enabled on the *Brocade* device. TCP port 80 (HTTP) allows access to the device's Web management interface.

By default, TCP port 280 for HP Top tools is disabled. This tool allows access to the device by HP ProCurve Manager.

The **no web-management** command disables both TCP ports. However, if you want to disable only port 280 and leave port 80 enabled, use the **hp-top-tools** option with the command. Here is an example.

```
BigIron RX(config)# no web-management hp-top-tools
```

Syntax: [no] web-management hp-top-tools

The **hp-top-tools** parameter disables TCP port 280.

Disabling SNMP access

SNMP is enabled by default on the device. SNMP is required if you want to manage a device using *IronView Network Manager*.

Enter the command to disable SNMP management of the device.

```
BigIron RX(config)#no snmp-server enable
```

Enter the command to later re-enable SNMP management of the device.

```
BigIron RX(config)#snmp-server enable
```

Syntax: [no] snmp-server enable

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 71.
- 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 72.

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 75.

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.

```
BigIron RX(config)# enable telnet password letmein
```

Syntax: [no] enable telnet password <string>

Suppressing Telnet connection rejection messages

By default, if a device 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 device. 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.

```
BigIron RX(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 75.

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 109.

Follow the steps 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.

```
BigIron RX> enable
BigIron RX#
```

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

```
BigIron RX# configure terminal
BigIron RX(config)#
```

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

```
BigIron RX(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.

```
BigIron RX(config)# enable port-config-password <text>
BigIron RX(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 74.

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 User EXEC and Privileged EXEC levels
  - The port-specific parts of the CONFIG level
  - All interface configuration levels
- Read Only level gives access to:
  - 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.

```
BigIron RX(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, BigIron RX> or BigIron RX#
- **configure** – CONFIG level; for example, BigIron RX`(config)#`
- **interface** – Interface level; for example, BigIron RX`(config-if-e10000-6)#`
- **virtual-interface** – Virtual-interface level; for example, BigIron RX`(config-vif-6)#`
- **rip-router** – RIP router level; for example, BigIron RX`(config-rip-router)#`
- **ospf-router** – OSPF router level; for example, BigIron RX`(config-ospf-router)#`
- **bgp-router** – BGP4 router level; for example, BigIron RX`(config-bgp-router)#`
- **port-vlan** – Port-based VLAN level; for example, BigIron RX`(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.

Follow the steps 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.
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.

```
BigIron RX(config)# enable password-display
BigIron RX(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 *Brocade* 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.

```
BigIron RX(config)# no service password-encryption
```

Syntax: [no] service password-encryption

Specifying a minimum password length

By default, the *Brocade* 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.

```
BigIron RX(config)# enable password-min-length 8
```

Syntax: 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 device. User accounts regulate who can access the management functions in the CLI using the following methods:

- Telnet access
- Web Management access
- SNMP access

Local user accounts provide greater flexibility for controlling management access to the device 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 72.

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 109.

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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config)# show users
```

Syntax: show users

Changing local user passwords

This section shows how to change the password for an existing local user account.The device 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".

Using the CLI

To change a local user password using the CLI, enter a command such as the following at the global CONFIG level of the CLI.

NOTE
You must be logged on with Super User access (privilege level 0) to change user passwords.

BigIron(config)# username wonka password willy

This command changes wonka's user name password to “willy”.

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

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

The <password-string> parameter is the user password. The password can be up to 255 characters and must differ from the current password and two previously configured passwords.

Using the Web Management Interface

To change a local user password using the Web Management Interface, you must first delete the user account, then re-add it with the new password. Use the following procedure.

NOTE
Before you can change a local user account using the Web Management Interface, you must enable this capability by entering the CLI command "password-change any" at the global CONFIG level of the CLI.

1. Log in to the Web Management Interface using a valid user name and password that has a read-write privilege level.
2. Select *Configure->System->Management->User Account*.
3. User account information is listed in a table. Click on the **Delete** button next to the user account whose password you wish to change.
4. Click on **Add User Account**.
5. Enter the user name in the **Username** field. The name cannot contain blanks.
6. Enter the password in the **Password** field. The password cannot contain blanks.

7. If necessary, select the management privilege level from the Privilege pulldown menu. By default, the system assigns privilege level 5 (Read-Only), which allows the user to display information but not to make configuration changes.
8. Click the **Add** button to save the change to the device's running-config file.
9. Repeat step 3 to step 8 for each user account.
10. Select the **Save** link at the bottom of the dialog. Select **Yes** when prompted to save the configuration.

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

Example

```
BigIron RX# 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.

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 device. 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 device
- Importing an RSA certificate and private key file from a client (optional)
- Generating a certificate

Enabling the SSL server on the device

To enable the SSL server on the device, enter the following command.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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 device 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 connecting client to the server. 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 *Brocade* device to create them.

If you want to allow the *Brocade* 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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config)# crypto-ssl certificate generate
```

Syntax: [no] crypto-ssl certificate generate

Deleting the SSL certificate

To delete the SSL certificate, enter the following command.

```
BigIron RX(config)# crypto-ssl certificate zeroize
```

Syntax: [no] crypto-ssl certificate zeroize

Configuring TACACS/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 device:

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

NOTE
You cannot authenticate *IronView Network Manager* (SNMP) access to a device using TACACS/TACACS+.

The TACACS and TACACS+ protocols define how authentication, authorization, and accounting information is sent between a device and an authentication database on a TACACS/TACACS+ server. TACACS/TACACS+ services are maintained in a database, typically on a UNIX workstation or PC with a TACACS/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 device and the TACACS+ server. TACACS+ allows for arbitrary length and content authentication exchanges, which allow any authentication mechanism to be utilized with the device. TACACS+ is extensible to provide for site customization and future development features. The protocol allows the device 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/TACACS+ authentication, authorization, and accounting

When you configure a device to use a TACACS/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/TACACS+ server.

If you are using TACACS+, *Brocade* recommends that you also configure ***authorization***, in which the device 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 device 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.

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 89.

TACACS authentication

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

When TACACS authentication takes place, the following events occur.

1. A user attempts to gain access to the device by doing one of the following:
   - Logging into the device using 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 device 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

When TACACS+ authentication takes place, the following events occur.

1. A user attempts to gain access to the device by doing one of the following:
   - Logging into the device using 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 device obtains a password prompt from a TACACS+ server.
5. The user is prompted for a password.
6. The user enters a password.
7. The device 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 device supports 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.

When TACACS+ exec authorization takes place, the following events occur.

1. A user logs into the device using Telnet, SSH, or the Web Management Interface
2. The user is authenticated.
3. The device 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.

When TACACS+ command authorization takes place, the following events occur.

1. A Telnet, SSH, or Web Management Interface user previously authenticated by a TACACS+ server enters a command on the device.
2. The device 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 device 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

TACACS+ accounting works as follows.

1. One of the following events occur on the device:
   - 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 device checks its configuration to see if the event is one for which TACACS+ accounting is required.
3. If the event requires TACACS+ accounting, the device 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 device sends an Accounting Stop packet to the TACACS+ accounting server.
7. The TACACS+ accounting server acknowledges the Accounting Stop packet.

AAA operations for TACACS/TACACS+

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a device that has TACACS/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 Telnet/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 Telnet/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 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 action	Applicable AAA operations
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 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.

TACACS/TACACS+ configuration considerations

Consider the following before you configure TACACS/TACACS+:

- You must deploy at least one TACACS/TACACS+ server in your network.
- The device supports authentication using up to eight TACACS/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 *Brocade* device to authenticate using a TACACS or TACACS+ server, not both.

TACACS configuration procedure

For TACACS configurations, use the following procedure.

1. Identify TACACS servers. Refer to "Identifying the TACACS/TACACS+ servers" on page 85.
2. Set optional parameters. Refer to "Setting optional TACACS/TACACS+ parameters" on page 86.
3. Configure authentication-method lists. Refer to "Configuring authentication-method lists for TACACS/TACACS+" on page 88.

TACACS+ configuration procedure

For TACACS+ configurations, use the following procedure.

1. Enable TACACS, refer to "Enabling SNMP to configure TACACS/TACACS" on page 85
2. Identify TACACS+ servers. Refer to "Identifying the TACACS/TACACS+ servers" on page 85.

3. Set optional parameters. Refer to "Setting optional TACACS/TACACS+ parameters" on page 86.
4. Configure authentication-method lists. Refer to "Configuring authentication-method lists for TACACS/TACACS+" on page 88.
5. Optionally configure TACACS+ authorization. Refer to "Configuring TACACS+ authorization" on page 89.
6. Optionally configure TACACS+ accounting. Refer to "Configuring TACACS+ accounting" on page 92.

Enabling SNMP to configure TACACS/TACACS

TACACS is disabled by default. To enable SNMP access to TACACS MIB objects on the device, enter the following command.

```
BigIron RX(config)#enable snmp config-tacacs
```

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

The <config-radius> parameter specifies the RADIUS configuration mode. Radius is disabled by default.

The <config-tacacs> parameter specifies the TACACS configuration mode. TACACS is disabled by default.

Identifying the TACACS/TACACS+ servers

To use TACACS/TACACS+ servers to authenticate access to a device, you must identify the servers to the device.

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

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

Syntax: 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/TACACS+ authentication servers to the device, 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/TACACS+ server by entering **no** followed by the **tacacs-server** command. For example, to remove 207.94.6.161, enter the following command.

```
BigIron RX(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/TACACS+** as an authentication method. (Refer to "Configuring authentication-method lists for TACACS/TACACS+" on page 88.) Otherwise, when you exit from the CONFIG mode or from a Telnet session, the system continues to believe it is TACACS/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 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.

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

Syntax: tacacs-server host <ip-addr> | <server-name> [auth-port <number> [authentication-only | authorization-only | accounting-only | default] [key <string>]]

The **default** parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization or accounting. 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 a server that can perform the requested function is found, or every server in the configured list has been tried.

Setting optional TACACS/TACACS+ parameters

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

- **TACACS+ key** – This parameter specifies the value that the *Brocade* device sends to the TACACS+ server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the *Brocade* device will resend an authentication request when the TACACS/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 *Brocade* 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 *Brocade* device waits for a response from a TACACS/TACACS+ server before either retrying the authentication request, or determining that the TACACS/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 device should match the one configured on the TACACS+ server. The key can be from 1 – 32 characters in length 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 device.

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

```
BigIron RX(config)# tacacs-server key rkwong
```

Syntax: tacacs-server key [0 | 1] <string>

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

Example

```
BigIron RX(config)# tacacs-server key 1 abc
BigIron RX(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 device will resend an authentication request when the TACACS/TACACS+ server does not respond. The retransmit limit can be from 1 – 5 times. The default is 3 times.

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

```
BigIron RX(config)# tacacs-server retransmit 5
```

Syntax: tacacs-server retransmit <number>

Setting the dead time parameter

The **dead-time** parameter specifies how long the 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.

To set the TACACS/TACACS+ dead-time value, enter the following command.

```
BigIron RX(config)# tacacs-server dead-time 5
```

Syntax: tacacs-server dead-time <number>

Setting the timeout parameter

The **timeout** parameter specifies how many seconds the *Brocade* device waits for a response from the TACACS/TACACS+ server before either retrying the authentication request, or determining that the TACACS/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.

```
BigIron RX(config)# tacacs-server timeout 5
```

Syntax: tacacs-server timeout <number>

Configuring authentication-method lists for TACACS/TACACS+

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

Within the authentication-method list, TACACS/TACACS+ is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If TACACS/TACACS+ 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 TACACS/TACACS+ authentication, you must create a separate authentication-method list for Telnet/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/TACACS+ as the primary authentication method for securing Telnet/SSH access to the CLI.

```
BigIron RX(config)# enable telnet authentication
BigIron RX(config)# aaa authentication login default tacacs local
```

The commands above cause TACACS/TACACS+ to be the primary authentication method for securing Telnet/SSH access to the CLI. If TACACS/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/TACACS+ as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI.

```
BigIron RX(config)# aaa authentication enable default tacacs local none
```

The command above causes TACACS/TACACS+ to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If TACACS/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 "Examples of authentication-method lists" on page 111.

NOTE
For examples of how to define authentication-method lists for types of authentication other than TACACS/TACACS+, refer to "Configuring authentication-method lists" on page 109.

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. Optionally, 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.

```
BigIron RX(config)# aaa authentication login privilege-mode
```

Syntax: 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 (up to 255 characters) 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 *Brocade* device to prompt only for a password. The device uses the username 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 `device` 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.

```
BigIron RX(config)# aaa authentication enable implicit-user
```

Syntax: [no] aaa authentication enable implicit-user

Telnet/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 device 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 device consults a TACACS+ server to determine the privilege level of the authenticated user.

To configure TACACS+ exec authorization on the device, enter the following command.

```
BigIron RX(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 "foundryprivlvl" 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 privilege-mode** command must also exist in the configuration.

Configuring an Attribute-Value pair on the TACACS+ server

During TACACS+ exec authorization, the *Brocade* 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 device 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 device extracts the last A-V pair configured for the Exec service that has a numeric value. The device 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 {
     privlvl = 15
        }
}
```

The attribute name in the A-V pair is not significant; the device uses the last one that has a numeric value. However, the device 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 deviceassociates a value from a non-"foundry-privlvl" A-V pair with a *Brocade* privilege level.

TABLE 34 Foundry equivalents for non-"foundry-privlvl" A-V pair values

Value for non-"foundry-privlvl" A-V pair	Foundry 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 `privlvl = 15`. The device 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-"foundry-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
     privlvl = 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 device.

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

Configuring command authorization

When TACACS+ command authorization is enabled, the device 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 device to perform authorization for the commands available at the Super User privilege level (that is, all commands on the device), enter the following command.

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

Syntax: 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 *IronView Network Manager* .

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

- **At all level**s: **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.

AAA support for console commands

To enable AAA support for commands entered at the console, enter the following command.

```
BigIron RX(config)# enable aaa console
```

Syntax: [no] enable aaa console

NOTES: AAA support for commands entered at the console can include the following:

- Login prompt that uses AAA authentication, using authentication-method lists
- Exec Authorization
- Exec Accounting
- System Accounting

Configuring TACACS+ accounting

The device supports TACACS+ accounting for recording information about user activity and system events. When you configure TACACS+ accounting on a BigIron RX, 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/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 device, and an Accounting Stop packet when the user logs out.

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

Syntax: 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 device 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.

```
BigIron RX(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: 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 TACACS+ accounting for system events

You can configure TACACS+ accounting to record when system events occur on the device. 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.

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

Syntax: aaa accounting system default start-stop radius | tacacs+ | none

Configuring an interface as the source for all TACACS/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/TACACS+ packets from the device. Identifying a single source IP address for TACACS/TACACS+ packets provides the following benefits:

- If your TACACS/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/TACACS+ server by configuring the *Brocade* device to always send the TACACS/TACACS+ packets from the same link or source address.
- If you specify a loopback interface as the single source for TACACS/TACACS+ packets, TACACS/TACACS+ servers can receive the packets regardless of the states of individual links. Thus, if a link to the TACACS/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/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/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/TACACS+ packets originated by the device.

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

```
BigIron RX(config)# int ve 1
BigIron RX(config-vif-1)# ip address 10.0.0.3/24
BigIron RX(config-vif-1)# exit
BigIron RX(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/TACACS+ packets from the device.

Syntax: 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/TACACS+ statistics and configuration information

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

Example

```
BigIron RX# show aaa
Tacacs+ key: foundry
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/TACACS+ information displayed by the **show aaa** command.

TABLE 35 Output of the show aaa command for TACACS/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.
Tacacs+ dead-time	The setting configured with the **tacacs-server dead-time** command.
Tacacs+ Server	For each TACACS/TACACS+ server, the IP address, port, and the following statistics are displayed: opensNumber of times the port was opened for communication with the server closesNumber of times the port was closed normally timeoutsNumber of times port was closed due to a timeout errorsNumber of times an error occurred while opening the port packets inNumber of packets received from the server packets outNumber 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

```
BigIron RX(config)#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 device:

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

NOTE
The device does not support RADIUS security for SNMP (*IronView Network Manager*) access.

RADIUS authentication, authorization, and accounting

When RADIUS authentication is implemented, the device consults a RADIUS server to verify user names and passwords. You can optionally configure RADIUS **authorization**, in which the device consults a list of commands supplied by the RADIUS server to determine whether a user can execute a command he or she has entered, as well as **accounting**, which causes the device 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.

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 104.

RADIUS authentication

When RADIUS authentication takes place, the following events occur.

1. A user attempts to gain access to the device by doing one of the following:
   - Logging into the device using 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 device sends a RADIUS Access-Request packet containing the username and password to the RADIUS server.

5. The RADIUS server validates the 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, the RADIUS server sends an Access-Accept packet to the device, authenticating the user. Within the Access-Accept packet are three *Brocade* 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.
9. The user is authenticated, and the information supplied in the Access-Accept packet for the user is stored on the device. 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.

RADIUS authorization

When RADIUS authorization takes place, the following events occur.

1. A user previously authenticated by a RADIUS server enters a command on the device.
2. The device 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 device 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 BigIron RX. 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 BigIron RX.
4. If the command list indicates that the user is authorized to use the command, the command is executed.

RADIUS accounting

RADIUS accounting works as follows.

1. One of the following events occur on the device:
   - 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 device checks its configuration to see if the event is one for which RADIUS accounting is required.

3. If the event requires RADIUS accounting, the device 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 device 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 BigIron RX 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/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 logs out of Telnet/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 action	Applicable AAA operations
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 to configure RADIUS:

- You must deploy at least one RADIUS server in your network.
- The device 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 *Brocade* 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 BigIron RX for RADIUS.

1. Configure *Brocade* vendor-specific attributes on the RADIUS server. Refer to "Configuring Brocade-specific attributes on the RADIUS server" on page 100.
2. Identify the RADIUS server to the device. Refer to "Identifying the RADIUS server to the BigIron RX" on page 101.
3. Set RADIUS parameters. Refer to "Setting RADIUS parameters" on page 102.

4. Configure authentication-method lists. Refer to "Configuring authentication-method lists for RADIUS" on page 103.
5. Optionally configure RADIUS authorization. Refer to "Configuring RADIUS authorization" on page 104.
6. Optionally configure RADIUS accounting. "Configuring RADIUS accounting" on page 106.

Configuring *Brocade*-specific attributes on the RADIUS server

NOTE
For the device, RADIUS Challenge is supported for 802.1x authentication but not for login authentication.

During the RADIUS authentication process, if a user supplies a valid username and password, the RADIUS server sends an Access-Accept packet to the device, authenticating the user. Within the Access-Accept packet are three *Brocade* 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

You must add these three *Brocade* vendor-specific attributes to your RADIUS server's configuration, and configure the attributes in the individual or group profiles of the users that will access the device.

Brocade's Vendor-ID is 1991, with Vendor-Type 1. The following table describes the *Brocade* vendor-specific attributes.

TABLE 36 ***Brocade*** **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.

TABLE 36 *Brocade* vendor-specific attributes for RADIUS (Continued)

Attribute name	Attribute ID	Data type	Description
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*
foundry-command-exception-flag	3	integer	Specifies whether the commands indicated by the foundry-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 foundry-command-string, deny all other commands. 1 Deny execution of the commands indicated by foundry-command-string, permit all other commands.

Enabling SNMP to configure RADIUS

RADIUS is disabled by default. To enable SNMP access to RADIUS MIB objects on the device, enter a command such as the following.

```
BigIron RX(config)#enable snmp config-radius
```

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

The <config-radius> parameter specifies the RADIUS configuration mode. RADIUS is disabled by default.

The <config-tacacs> parameter specifies the TACACS configuration mode. TACACS is disabled by default.

Identifying the RADIUS server to the BigIron RX

To use a RADIUS server to authenticate access to a BigIron RX, you must identify the server to the device.

Example

```
BigIron RX(config)# radius-server host 209.157.22.99
```

Syntax: 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, authorization, and accounting.

```
BigIron RX(config)# radius-server host 1.2.3.4 authentication-only key abc
BigIron RX(config)# radius-server host 1.2.3.5 authorization-only key def
BigIron RX(config)# radius-server host 1.2.3.6 accounting-only key ghi
```

Syntax: radius-server host <ip-addr> | <server-name> [auth-port <number> acct-port <number> [authentication-only | authorization-only | accounting-only | default] [key <string>]]

The **default** parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization or accounting. 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 a server that can perform the requested function is found, or every server in the configured list has been tried.

Setting RADIUS parameters

You can set the following parameters in a RADIUS configuration:

- **RADIUS key** – This parameter specifies the value that the device sends to the RADIUS server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the device 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 device 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 device should match the one configured on the RADIUS server. The key can be from 1 – 32 characters in length and cannot include any space characters.

Use the command to specify a RADIUS server key.

```
BigIron RX(config)# radius-server key mirabeau
```

Syntax: radius-server key [0 | 1] <string>

When you display the configuration of the BigIron RX, the RADIUS key is encrypted.

Example

```
BigIron RX(config)# radius-server key 1 abc
BigIron RX(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 *Brocade* 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.

Use the command to set the RADIUS retransmit limit.

```
BigIron RX(config)# radius-server retransmit 5
```

Syntax: radius-server retransmit <number>

Setting the timeout parameter

The **timeout** parameter specifies how many seconds the BigIron RX 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.

```
BigIron RX(config)# radius-server timeout 5
```

Syntax: radius-server timeout <number>

Configuring authentication-method lists for RADIUS

You can use RADIUS to authenticate Telnet/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.

```
BigIron RX(config)# enable telnet authentication
BigIron RX(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.

```
BigIron RX(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 "Examples of authentication-method lists" on page 111.

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 109.

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.

```
BigIron RX(config)# aaa authentication login privilege-mode
```

Syntax: 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 device 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 device 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.

```
BigIron RX(config)# aaa authentication enable implicit-user
```

Syntax: [no] aaa authentication enable implicit-user

Configuring RADIUS authorization

The device 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 the BigIron RX, make sure that the **aaa authentication enable default radius** command or the **aaa authentication login privilege-mode** command exist in the configuration.

When RADIUS exec authorization is performed, the device consults a RADIUS server to determine the privilege level of the authenticated user.

To configure RADIUS exec authorization on the device, enter the following command.

```
BigIron RX(config)# aaa authentication exec default radius
```

Syntax: aaa authentication 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 foundry-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 foundry-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 enable default radius** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring command authorization

When RADIUS command authorization is enabled, the device 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 device to perform authorization for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
BigIron RX(config)# aaa authorization commands 0 default radius
```

Syntax: 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 device 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 *IronView Network Manager*, .

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 device supports 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 command.

```
BigIron RX(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 device supports RADIUS accounting for recording information about user activity and system events. When you configure RADIUS accounting on device, 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/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 device, and an Accounting Stop packet when the user logs out.

```
BigIron RX(config)# aaa accounting exec default start-stop radius
```

Syntax: 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 the device 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.

```
BigIron RX(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: 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 the device. 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.

```
BigIron RX(config)# aaa accounting system default start-stop radius
```

Syntax: 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 device. 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 device 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/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.

```
BigIron RX(config)# int ve 1
BigIron RX(config-vif-1)# ip address 10.0.0.3/24
BigIron RX(config-vif-1)# exit
BigIron RX(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 device.

Syntax: 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/TACACS+ and RADIUS servers identified on the device.

Example

```
BigIron RX# show aaa
Tacacs+ key: foundry
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 37 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 PortRADIUS authentication port number (default 1645) Acct PortRADIUS accounting port number (default 1646) opensNumber of times the port was opened for communication with the server closesNumber of times the port was closed normally timeoutsNumber of times port was closed due to a timeout errorsNumber of times an error occurred while opening the port packets inNumber of packets received from the server packets outNumber 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

```
BigIron RX(config)# show web
User                              Privilege       IP address
set                                      0       192.168.1.234
```

Syntax: show web

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/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 63 or "Restricting remote access to the device to specific IP addresses" on page 66.

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

Consider the following before configuring authentication-method lists:

- 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/TACACS+ security" on page 80.

- 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 *IronView Network Manager*, 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 *IronView Network Manager*, configure an authentication-method list for SNMP access and specify “local” as the primary authentication method.

Examples of authentication-method lists

Example

The following example shows how to configure authentication-method lists for the Web Management Interface, *IronView Network Manager*, 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.

```
BigIron RX(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 *IronView Network Manager*, enter a command such as the following.

```
BigIron RX(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 *IronView Network Manager*.

To configure an authentication-method list for the Privileged EXEC and CONFIG levels of the CLI, enter the following command.

```
BigIron RX(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.

```
BigIron RX(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 *Brocade* device authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web management interface.

NOTE
TACACS/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 38.

TABLE 38 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 71.
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 72.
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 76.
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.

Chapter

5

Configuring Basic Parameters

In this chapter

This chapter describes how to configure basic system parameters.

The device 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
For information about the Syslog buffer and messages, refer to Appendix A, “Using Syslog”.

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 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, contact, and location, enter commands such as the following.

```
BigIron RX(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 *BigIron RX*.

Syntax: hostname <string>

Syntax: snmp-server contact <string>

Syntax: snmp-server location <string>

The name, contact, and location each can be up to 32 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.

NOTE
The **chassis name** command does not change the CLI prompt. Instead, the command assigns an administrative ID to the device.

Configuring Simple Network Management Protocol(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 device 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/TACACS+ server.

NOTE
To add and modify “get” (read-only) and “set” (read-write) community strings, refer to Chapter 4, “Securing Access to Management Functions”.

Specifying an SNMP trap receiver

You can specify a trap receiver to ensure that all SNMP traps sent by the device 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 device sends all the SNMP traps to the specified hosts and includes the specified community string. Administrators can therefore filter for traps from a device based on IP address or community string.

When you add a trap receiver, you can specify whether to have the community string encrypted or to have it shown in the clear. In either case, the software does not encrypt the string in the SNMP traps sent to the receiver.

To specify an SNMP trap receiver, enter a command such as the following.

```
BigIron RX(config)# snmp-server host 2.2.2.2 1 mypublic port 200
BigIron RX(config)# write memory
```

The first commands adds trap receiver 2.2.2.2, configures the software to encrypt display of the community string, and designates the UDP port that will be used to receive traps. The second command saves the community string to the startup configuration file, and the software adds the following command to the file.

```
snmp-server host 2.2.2.2 1 <encrypted-string>
```

Syntax: snmp-server host <ip-addr> [0 | 1] <string> [port <value>]

The <ip-addr> parameter specifies the IP address of the trap receiver.

The **0 | 1** parameter specifies whether you want the software to encrypt the string (**1**) or show the string in the clear (**0**). The default is 0.

The <string> parameter specifies an SNMP community string configured on the device. It can be a read-only string or a read-write string. 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 device 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.

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, *IronView Network Manager* and another network management application can coexist in the same system. The device 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 device 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 device 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 device 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 device to send all SNMP traps from the first configured IP address on port 4/11, enter the following commands.

```
BigIron RX(config)# snmp-server trap-source ethernet 4/11
BigIron RX(config)# write memory
```

Syntax: snmp-server trap-source loopback <num> | ethernet <slot/port> | ve <num>

The <num> parameter is a loopback interface or virtual routing interface number.

To specify a loopback interface as the device's SNMP trap source, enter commands such as the following.

```
BigIron RX(config)# int loopback 1
BigIron RX(config-lbif-1)# ip address 10.0.0.1/24
BigIron RX(config-lbif-1)# exit
BigIron RX(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 device. Regardless of the port the device 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 device 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 device might not be able to reach the servers, in which case the messages are lost.

By default, the device 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 device 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.

```
BigIron RX(config)# snmp-server enable traps holddown-time 30
```

The command changes the holddown time for SNMP traps to 30 seconds. The device 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 device 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
- Power supply failure
- Fan failure
- Cold start
- Link up
- Link down
- Bridge new root
- Bridge topology change
- Locked address violation
- Module insert
- Module remove
- BGP4
- OSPF
- FSRP
- VRRP
- VRRPE

To stop link down occurrences from being reported, enter the following.

```
BigIron RX(config)# no snmp-server enable traps link-down
```

Syntax: [no] snmp-server enable traps <trap-type>

A list of *Brocade* traps is available in the *MIB Reference Guide*.

Disabling Syslog messages and traps for CLI access

The device 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 authenticated by an authentication-method list based on a local user account, RADIUS server, or TACACS/TACACS+ server.

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/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.

```
BigIron RX(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.

Disabling the Syslog messages and traps

Logging of CLI access is enabled by default. To disable logging of CLI access, enter the following commands.

```
BigIron RX(config)# no logging enable user-login
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# reload
```

Syntax: [no] logging enable user-login

Refer to the MIB Guide for a list of traps.

Configuring an interface as the source for all Telnet packets

You can designate the lowest-numbered IP address configured an interface as the source IP address for all Telnet packets from the device. 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 device 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.

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

The software uses the lowest-numbered IP address configured on the interface as the source IP address for Telnet packets originated by the device.

To specify the lowest-numbered IP address configured on a virtual routing interface as the device's source for all Telnet packets, enter commands such as the following.

```
BigIron RX(config)# int loopback 2
BigIron RX(config-lbif-2)# ip address 10.0.0.2/24
BigIron RX(config-lbif-2)# exit
BigIron RX(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 device.

Syntax: 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 device.

```
BigIron RX(config)# interface ethernet 1/4
BigIron RX(config-if-e10000-1/4)# ip address 209.157.22.110/24
BigIron RX(config-if-e10000-1/4)# exit
BigIron RX(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 doing the following.

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 TFTP packets

You can configure the device 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. The software uses the lowest-numbered IP 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 routing interface as the device's source for all TFTP packets, enter commands such as the following.

```
BigIron RX(config)# int ve 1
BigIron RX(config-vif-1)# ip address 10.0.0.3/24
BigIron RX(config-vif-1)# exit
BigIron RX(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 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.

```
BigIron RX(config)# int ve 1
BigIron RX(config-vif-1)# ip address 10.0.0.4/24
BigIron RX(config-vif-1)# exit
BigIron RX(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.

NOTE
With this new command, the source ip of syslog is no longer controlled by the snmp-server trap-source command. In releases berfore 02.4.00, the **snmp-server trap-source** command controlled both SNMP and Syslog source IP.

Specifying a Simple Network Time Protocol (SNTP) server

You can configure the device to consult SNTP servers for the current system time and date.

NOTE
The device does not retain time and date information across power cycles. Unless you want to reconfigure the system time counter each time the system is reset, Brocade recommends that you use the SNTP feature.

To identify an SNTP server with IP address 208.99.8.95 to act as the clock reference for a device, enter the following.

```
BigIron RX(config)# sntp server 208.99.8.95
```

Syntax: sntp server <ip-addr> | <hostname> [<version>]

The <version> parameter specifies the SNTP version the server is running and can be from 1 – 4. The default is 1. You can configure up to three SNTP servers by entering three separate **sntp server** commands.

By default, the device polls its SNTP server every 30 minutes (1800 seconds). To configure the device to poll for clock updates from a SNTP server every 15 minutes, enter the following.

```
BigIron RX(config)# sntp poll-interval 900
```

Syntax: [no] sntp poll-interval <1-65535>

To display information about SNTP associations, enter the following command.

```
BigIron RX# show sntp associations
  address         ref clock       st  when  poll  delay  disp
 ~207.95.6.102    0.0.0.0         16   202     4    0.0     5.45
 ~207.95.6.101    0.0.0.0         16   202     0    0.0     0.0
* synced, ~ configured
```

Syntax: show sntp associations

The following table describes the information displayed by the **show sntp associations** command.

TABLE 39 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
st	NTP stratum level of the peer
when	Amount of time since the last NTP packet was received from the peer
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.

```
BigIron RX# show sntp status
Clock is unsynchronized, stratum = 0, no reference clock
precision is 2**0
reference time is 0       .0
clock offset is 0.0    msec, root delay is 0.0  msec
root dispersion is 0.0  msec, peer dispersion is 0.0  msec
```

Syntax: show sntp status

The following table describes the information displayed by the **show sntp status** command.

TABLE 40 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 this system
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

Setting the system clock

In addition to SNTP support, the device also allows you to set the system time counter. It starts the system time and date clock with the time and date you specify. The time counter setting is not retained across power cycles and 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, you will need to re-enter the time and date following each reboot.

For more details about SNTP, refer to "Specifying a Simple Network Time Protocol (SNTP) server" on page 121.

To set the system time and date to 10:15:05 on October 15, 2005, enter the following command.

```
BigIron RX# clock set 10:15:05 10-15-05
```

Syntax: [no] clock set <hh:mm:ss> <mm-dd-yy> | <mm-dd-yyyy>

By default, the device does not change the system time for daylight savings time. To enable daylight savings time, enter the following command.

```
BigIron RX# clock summer-time
```

Syntax: clock summer-time

Although SNTP servers typically deliver the time and date in Greenwich Mean Time (GMT), you can configure the device 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.

Beginning with the Multi-Service IronWare 02.8.01 release, you can now set the system time clock for countries like India that fall in the ½ hour time zone. Only the following zones have been added:

- GMT + 11:30
- GMT + 10:30
- GMT + 09:30
- GMT + 06:30
- GMT + 05:30
- GMT + 04:30
- GMT + 03:30
- GMT - 03:30
- GMT - 08:30
- GMT - 09:30

To change the time zone to Australian East Coast time (which is normally 10 hours ahead of GMT), enter the following command.

```
BigIron RX(config)# clock timezone gmt gmt+10
```

Syntax: 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.

New Daylight Saving Time (DST)

The new Daylight Saving Time (DST) change that went into effect on March 11th, 2007 affects only networks following the US time zones. This software release supports the DST automatic feature, but to trigger the device to the correct time, the device must be configured to the US time zone, not the GMT offset. To configure your device to use the US time zone, enter the following command.

```
BigIron RX(config)#clock timezone us pacific
```

Syntax: [no] clock timezone us <timezone-type>

Enter pacific, eastern, central, or mountain for <timezone-type>.

This command must be configured on every device that follows the US DST.

To verify the change, run a **show clock** command.

```
BigIron RX(config)#show clock
```

Syntax: show clock

Refer to *October 19, 2006 - Daylight Savings Time 2007* Advisory, posted on *kp.foundrynet.com* for more information.

Configuring CLI banners

The device 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 device 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 device 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 device!" when a Telnet CLI session is established.

```
BigIron RX(config)# banner motd $ (Press Return)
Enter TEXT message, End with the character '$'.
Welcome to device!! $
```

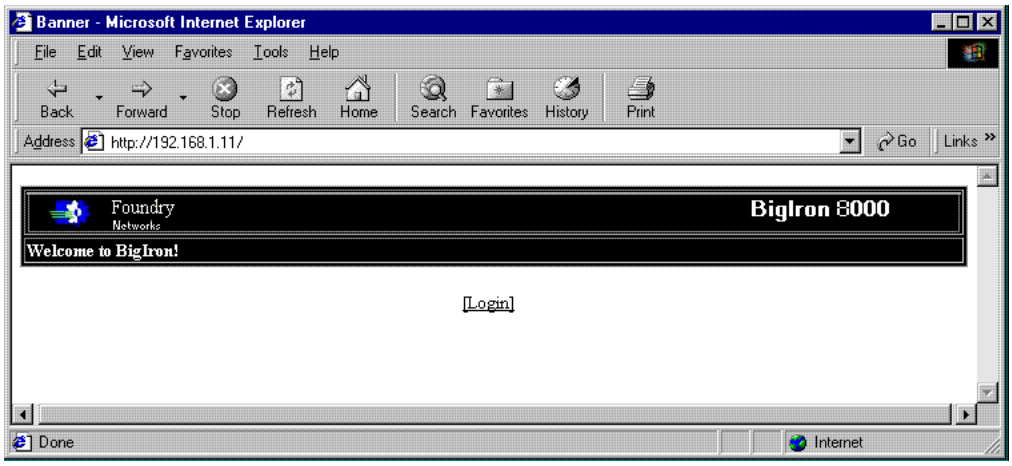
A delimiting character is established on the first line of the **banner motd** command. You begin and end the message with this delimiting character. The delimiting character can be any character except "(double-quotation mark) and cannot appear in the banner text. In this example, the delimiting character 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
If a message of the day(MOTD) is configured, the user will be required to press the Enter key before the the user can login.

NOTE
The **banner** <delimiting-character> command is equivalent to the **banner motd** <delimiting-character> command.

When you access the Web Management Interface, the banner is displayed.



Setting a privileged EXEC CLI level banner

You can configure the device to display a message when a user enters the Privileged EXEC CLI level.

Example

```
BigIron RX(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 device 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

```
BigIron RX(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.

```
BigIron RX(config)# terminal length 15
```

Syntax: 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.

```
BigIron RX(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 device supports the following protocols:

- BGP4
- DVMRP
- FSRP
- IP

- OSPF
- PIM
- RIP
- VRRP
- VRRPE

By default, IP routing is enabled on the device. All other protocols are disabled, so you must enable them to configure and use them.

NOTE
The following protocols require a system reset before the protocol will be active on the system: PIM, DVMRP, RIP, FSRP. To reset a system, enter the **reload** command at the privileged level of the CLI.

To enable a protocol on a device, enter **router** at the global CONFIG level, followed by the protocol to be enabled. The following example shows how to enable OSPF.

```
BigIron RX(config)# router ospf
BigIron RX(config)# end
BigIron RX# write memory
BigIron RX# reload
```

Syntax: router bgp | dvmrp | ospf | pim | rip | vrrp | vrrpe

Displaying and modifying system parameter default settings

The device 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
- Layer 2 Port VLANs supported on a system
- Layer 3 Protocol VLANs supported on a system
- Layer 4 sessions supported
- IP cache size
- ARP entries
- IP routes
- IP route filters
- IP subnets per port and per device
- Static routes

The tables you can configure as well the defaults and valid ranges for each table differ depending on the device 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 device, 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, their defaults and maximum values, enter the following command at any level of the CLI.

```
BigIron RX# show default values
telnet@ro(config)#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.:140 sec   igmp query:60 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

System Parameters    Default    Maximum    Current
mac                  32768      65536      65536
vlan                 512        4095       512
spanning-tree        32         128        32
rstp                 32         128        32
ip-arp               8192       65536      8192
ip-static-arp        2048       4096       2048
multicast-route      8192       153600     8192
dvmrp-route          2048       16384      2048
dvmrp-mcache         4096       4096       4096
pim-mcache           4096       4096       4096
igmp-max-group-addr  1024       4096       1024
ip-cache             204800     524288     524288
ip-route             204800     524288     524288
ip-subnet-port       24         128        24
virtual-interface    255        4095       255
session-limit        32768      163840     32768
ip-filter-sys        4096       8192       4096
mgmt-port-acl-size   20         100        20
l2-acl-table-entries 64         256        64
vlan-multicast-flood 0          4095       0
ipv6-cache           65536      65536      65536
ipv6-route           65536      65536      65536
```

Syntax: show default values

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. For example, to increase the capacity of the IP route table, enter the following commands.

```
BigIron RX(config)# system-max ip-route 120000
BigIron RX(config)# write memory
BigIron RX(config)# exit
BigIron RX# 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 device from 24 to 64, then increase the total number of IP interfaces you can configure from 256 to 512, enter the following commands.

```
BigIron RX(config)# system-max subnet-per-interface 64
BigIron RX(config)# write memory
BigIron RX(config)# exit
BigIron RX# reload
```

Syntax: system-max subnet-per-interface <num>

The <num> parameter specifies the maximum number of subnet addresses per port and can be from 1 – 64. The default is 24.

Syntax: system-max subnet-per-system <num>

The <num> parameter specifies the maximum number of subnet addresses for the entire device and can be from 1 – 512. The default is 256.

```
BigIron RX(config)# system-max subnet-per-system 512
BigIron RX(config)# write memory
BigIron RX(config)# exit
BigIron RX# reload
```

To increase the size of the IP route table for static routes, enter the following command.

```
BigIron RX(config)# system-max ip-static-route 8192
```

Syntax: system-max ip-static-route <num>

The maximum number of static routes you can define is 4096.

NOTE
You must reload the software for the change to take effect.

Enabling or disabling Layer 2 switching

By default, *Brocade* device supports Layer 2 switching and switches the routing protocols that are not supported. You can disable Layer 2 switching globally or on individual ports.

NOTE
Make sure you really want to disable all Layer 2 switching operations before actually disabling it. Consult your reseller or Brocade for information.

To globally disable Layer 2 switching on the device, enter commands such as the following.

```
BigIron RX(config)# route-only
BigIron RX(config)# exit
BigIron RX# write memory
BigIron RX# reload
```

To re-enable Layer 2 switching globally, enter the following.

```
BigIron RX(config)# no route-only
BigIron RX(config)# exit
BigIron RX# write memory
BigIron RX# reload
```

Syntax: [no] route-only

To disable Layer 2 switching only on a specific interface, go to the Interface configuration level for that interface, then disable the feature. The following commands show how to disable Layer 2 switching on port 3/2.

```
BigIron RX(config)# interface ethernet 3/2
BigIron RX(config-if-e10000-3/2)# route-only
```

Syntax: [no] route-only

To re-enable Layer 2 switching, enter the command with “no”.

```
BigIron RX(config-if-e10000-3/2)# no route-only
```

CAM partitioning for the BigIron RX

In releases prior to 02.3.00, CAM partitioning was not configurable. Starting in BigIron RX software release 02.3.00, you can specify the percentage of CAM assigned to each of the CAM entry types globally. CAM Partitioning is not required on the device. The default CAM allocations are described in the following.

Resource limitation for BigIron RX hardware: MAC 16K and 512K IPv4 routes and 64K IPv6 routes simultaneously.

The number of CAM entries available for ACL, PBR, RL: 1024 Default values:

- **ACL** – 416
- **Rate Limiting** – 416 (Shared with PBR)

Other Default number of entries available:

- **IPv6 Multicast** – 192

Re-distributing CAM allocations

Depending on the needs of you network, the CAM allocations may need to be re-distributed. There are two steps to the command.

1. Change the allocation used between the rules + PBR/RL and the IPv6 multicast entries.
2. Change the allocation of the ACL rules and the PBR/RL entries.

The total amount of CAM entries available is 1024 for each packet processor. If you want to configure 600 for ACLs, 168 for PBR and Rate Limiters, and 256 for IPv6 multicast forwarding entries, enter commands such as the following.

```
BigIron RX(config)#cam-partition rw session 768
BigIron RX(config)#cam-partition rw session rule-partition 600
```

If you want to configure 2 ACL entries and 2 IPv6 entries and 1020 Rate Limiting entries, enter a command such as the following.

```
BigIron RX(config)#cam-partition rw session 1022
BigIron RX(config)#cam-partition rw session rule-partition 2
```

Syntax: cam-partition rw session <count>

Syntax: cam-partition rw session rule-partition <count>

The "cam-partition rw session xx" command allocates xx entries of the 1024 entries to ACLs+(RL/QoS/PBR) and 1024-xx to the IPv6 multicast entries.

The "cam-partition rw session rule-partition yy" command gives the yy of xx entries to ACLs and xx-yy entries to RL/QoS/PBR entries.

NOTE
A reload is required after a CAM partition command is configured for the CAM partition to take effect.

Nexthop table

The nexthop table on BigIron RX line cards contains next hop entries that are used by the routing table entries to route IP traffic. One or more routes can point to the same next hop. Entries for directly connected hosts are also present in the nexthop table. The nexthop table has 4096 entries per line card by default. This table is divided into four partitions. First partition contains next hop entries for routes with one routing path. This included directly connected host entries. Second partition contains next hop entries for routes with two or less equal cost paths. Allocation from this partition is always done in blocks of two entries. Third partition contains next hop entries for routes with four or less equal cost paths. Allocation from this partition is always done in blocks of four entries. The fourth partition contains next hop entries for routes with eight or less equal cost paths. Allocation from this partition is always done in blocks of eight entries. For a given route, the next hop entry is allocated based on the path count and best-fit algorithm. For example, to create a next hop entry for a route with three equal cost paths, the entries are allocated from the four-path partition. Four entries will be allocated from four-path partition even if the route has three equal cost paths. If the four-path partition is full, the entries are allocated from the next partition, which is the eight-path partition.

The Nexthop table is partitioned as follows:

- **One-path partition**: 2816 entries
- **Two-path partition**: 512 entries
- **Four-path partition**: 512 entries
- **Eight-path partition**: 256 entries

NOTE
A reload is required after a CAM partition command is configured for the CAM partition to take effect.

As of release 02.4.00, the Nexthop table is user configurable. If the router is installed in a network where there are many directly connected hosts, then the size of one-path partition should be increased. To configure the partition, use a command such as the following.

```
BigIron RX(config)# cam-partition next-hop 2048 1024 512 512
```

The above command partitions the next-hop table into 2048 one-path, 1024 two-path, 512 four-path and 512 eight-path entries. The two-path count must be a multiple of two. The four-path count must be a multiple of four and the eight-path count must be a multiple of eight. All values must be non-zero and the total must be 4096.

This command requires system reload. After the system reboots, the nexthop table on the line cards are partitioned according to the parameters in above command.

Syntax: cam-partition next-hop <number>

Use the <number> parameter to specify the number of entries for the nexthop.

Use the **no cam-partitioning next-hop** command to return to the default partitioning.

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.

```
BigIron RX(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 device automatically creates a static MAC entry.

NOTE
To delete the static MAC entry, you must delete the static ARP entry first.

For more information, refer to "IP fragmentation protection" on page 178 and "Creating static ARP entries" on page 183.

Chapter

Configuring Interface Parameters 6

In this chapter

Assigning a port name

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 322.

To configure trunk groups or dynamic link aggregation, refer to Chapter 8, "Link Aggregation".

All device 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.

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.

```
BigIron RX(config)# interface e 2/8
BigIron RX(config-if-e10000-2/8)# port-name Marsha Markey
```

Syntax: port-name <text>

The <text> parameter is an alphanumeric string. The name can be up to 255 characters long on the device. The name can contain 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.

```
BigIron RX(config)# interface e 1/8
BigIron RX(config)# ip address 192.45.6.110 255.255.255.0
```

Syntax: ip address <ip-addr> <ip-mask>

or

Syntax: ip address <ip-addr>/<mask-bits>

NOTE
You also can enter the IP address and mask in CIDR format, as follows:
BigIron RX`(config)# ip address 192.45.6.1/24`

Speed/Duplex negotiation

Speed/Duplex Negotiation detects the speed (10MBps, 100Mbps, 1000Mbps) and duplex (half-duplex or full-duplex) settings of the device on the other end of the wire and subsequently adjusts to match those settings.

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.

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.

The master and slave parameters are applicable only if the speed parameter is set to 1000. The value for this parameter must correspond to the value on the link partner--for example, if the local link has a value of master, the link partner must have a value of slave.

On 1000 Gigabit, auto-negotiation determines which side of the link is master and which side is slave.

NOTE
You can not force a full duplex speed on CuSFP when using two 24C's. Setting both sides to 100-full and using gig links or actual 24C's is recommended for switch uplinks. Host PC's that are connected are not affected.

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.

NOTE
Brocade recommends using gig links or 24C's links for switch uplinks when transmitting Layer 2 Traffic in a bidirectional patterns.

NOTE
To force the port to run at 1000 Mbps, set one of the link's ports to be the master for the link. To set a port as a Gigabit master port, enter the following command at the interface configuration level for the port:

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.

The following example configures the interface to 1000 Mbps, and designate it as the master port.

To force the port to run at 1000 Mbps, set one of the link's ports to be the master for the link. To set a port as a Gigabit master port, enter the following command at the interface configuration level for the port.

```
BigIron RX(config)#interface ethernet 1/5
BigIron RX(config-if-e10000-1/5)#speed-duplex 1000-master
```

The following example configures the interface to 1000 Mbps, and designate it as the slave port.

```
BigIron RX(config)#interface ethernet 2/4
BigIron RX(config-if-e10000-2/4)#speed-duplex 1000-slave
```

Syntax: [no] speed-duplex {auto |1000-master |1000-slave |1000-full | 100-full | 100-half | 10-full | 10-half}

auto - Autonegotiation

1000-master - Forces 1000 Mbps master port

1000-slave - Forces 1000 Mbps slave port

1000-full - Forces 1000 Mbps full-duplex operation

1000-half - Forces 100 Mbps half-duplex operation

100-full - Forces 100 Mbps full-duplex operation

100-half - Forces 100 Mbps half-duplex operation

10-full - Forces 10 Mbps full-duplex operation

10-half - Forces 10 Mbps half-duplex operation

Disabling or re-enabling a port

The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is enabled.

To disable port 8 on module 1 of a device, enter the following.

```
BigIron RX(config)# interface e 1/8
BigIron RX(config-if-e10000-1/8)# disable
```

Syntax: disable

Syntax: enable

You also can disable or re-enable a virtual routing interface. To do so, enter commands such as the following.

```
BigIron RX(config)# interface ve v1
BigIron RX(config-vif-1)# disable
```

Syntax: 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.

```
BigIron RX(config-vif-1)# enable
```

Syntax: enable

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

To change the mode for individual ports, enter commands such as the following.

```
BigIron RX(config)# int ethernet 4/1 to 4/4
BigIron RX(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: gig-default neg-full-auto | auto-gig | neg-off

Default is gig-default auto-gig

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). Flow control is enabled by default.

To disable flow control on full-duplex ports on a system, enter the following.

```
BigIron RX(config)# no flow-control
```

To turn the feature back on.

```
BigIron RX(config)# flow-control
```

Syntax: [no] 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 device 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
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 device, but can be disabled with the **no flow-control** command.

To specify threshold values for flow control, enter the following command.

```
BigIron RX(config)# qd-flow sink 75 sunk 50 slot 1
```

Syntax: 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 device 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 device drops traffic at the specified priority level.

The <slot> parameter specifies the location of the module where the thresholds are to take effect.

Locking a port to restrict addresses

Address-lock filters allow you to limit the number of devices that have access to a specific port. Access violations are reported as SNMP traps. By default this feature is disabled. A maximum of 2048 entries can be specified for access. The default address count is eight.

Example

To enable address locking for port 2/1 and place a limit of 15 entries.

```
BigIron RX(config)# lock e 2/1 addr 15
```

Syntax: lock-address ethernet <portnum> [addr-count <num>]

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.

This 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.

To have all ports come up at the same time during a system reload, enter a command such as the following.

```
BigIron RX(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.

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 would 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 trunk 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.

```
BigIron RX (config-if-e1000-1/2)# delay-link-event 2 down
```

Syntax: delay-link-event <time> up | down

The <time> parameter is the number of 50-ms units. The default is 0.

The <up> parameter means only "up" events are delayed.

The <down> parameter means that only the down events are delayed.

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 flap 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, the port's link state will remain disabled until it is manually re-enabled.

Configuration notes

- When a link dampening port becomes a member of a trunk group, that port, as well as all other member ports of that trunk group, will inherit the primary port's configuration. This means that the member ports will inherit the primary port's link dampening configuration, regardless of any previous configuration.
- The *Brocade* device counts the number of times a port's link state toggles from "up to down", and not from "down to up".
- The sampling time or window (the time during which the specified toggle threshold can occur before the wait period is activated) is triggered when the first "up to down" transition occurs.
- "Up to down" transitions include UDLD-based toggles, as well as the physical link state.

Configuring port flap dampening on an interface

This feature is configured at the interface level.

```
BigIron RX(config)# interface ethernet 2
BigIron RX(config-if-e100-2)# 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 value from 1 – 50.

NOTE
Brocade does not advise setting the <toggle-threshold> to a value lower than 2 for the following reason: When LACP is enabled on both the local and remote ends of a link, and port flap dampening is enabled on the local link, when the remote system forms a trunk group dynamically, it will disable then re-enable all member ports of that trunk group. The local system will detect and count this transition as an up-to-down toggle.

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 flap dampening on a trunk

You can configure the port flap dampening feature on the primary port of a trunk using the **link-error-disable** command. Once configured on the primary port, the feature is enabled on all ports that are members of the trunk. You cannot configure port flap dampening on port members of the trunk.

Enter commands such as the following on the primary port of a trunk.

```
BigIron RX(config)# interface ethernet 2
BigIron RX(config-if-e100-2)#link-error-disable 10 3 10
```

Re-enabling a port disabled by port flap dampening

A port disabled by port flap dampening is automatically re-enabled once the wait period expires; however, if the wait period is set to zero (0) seconds, you must re-enable the port by entering the following command on the disabled port.

```
BigIron RX(config)# interface ethernet 2
BigIron RX(config-if-e100-2)# no link-error-disable 10 3 10
```

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 Chapter 18, “Configuring Quality of Service”.

Assigning a mirror port and monitor ports

You can monitor traffic on *Brocade* 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.

On a 4 X 10G module, any port can operate as a mirror port and you can configure more than one mirror port. You can configure up to 64 mirror ports. You can configure the mirror ports on different modules and you can configure more than one mirror port on the same module.

Each mirror port can have its own set of monitored ports. For example, you can configure ports 1/1 and 5/1 as mirror ports, and monitor ports 1/2 – 1/8 on port 1/1 and ports 5/2 – 5/8 on port 5/1. The mirror port and monitored ports also can be on different slots.

However, on a 24 X 1G module, you can configure only one mirror port per packet processor (PPCR). For example, if you configure port 3/1 to be mirrored by port 5/1, all other ports that you want to be mirrored must use 5/1 as the mirror port. The following table shows which ports share the same PPCR.

Port numbers	PPCR
1 – 12	1
13 – 24	2

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.
- There can be only one mirror port per packet processor on a 24 X 1G module.
- For outbound traffic, there can be up to 8 active mirror ports system wide.

Configuring port mirroring and monitoring

You can configure multiple mirror ports on the same module. However, if you mirror inbound traffic to any of the mirror ports on the module, the traffic is mirrored to all the mirror ports on the module. If you plan to mirror outbound traffic only, you can use multiple mirror ports on the same module without the traffic being duplicated on the other mirror ports on the module.

NOTE
You cannot monitor outbound traffic from one armed router traffic.

NOTE
Mirror (analyzer) ports cannot be assigned to the 16x10 card. You can monitor traffic on 16x10 ports.

The following example configures two mirror ports on the same module and one mirror port on another module. It will illustrate how inbound traffic is mirrored to the two mirror ports on the same module even if the traffic is configured to be mirrored to only one mirror port on the module.

```
BigIron RX(config)# mirror-port ethernet 1/1
BigIron RX(config)# mirror-port ethernet 1/2
BigIron RX(config)# mirror-port ethernet 2/1
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e10000-3/1)# monitor ethernet 1/1 both
BigIron RX(config-if-e10000-3/1)# monitor ethernet 2/1 in
BigIron RX(config-if-e10000-3/1)# interface ethernet 4/13
BigIron RX(config-if-e10000-4/1)# monitor ethernet 1/2 both
```

This example configures two mirror ports 1/1 and 1/2 on the same module. It also configures input and output traffic from port 3/1 to be mirrored to mirror port 1/1 and input and output traffic from port 4/1 to be mirrored to mirror port 1/2. Because mirror ports 1/1 and 1/2 are configured on the same module, mirror port 1/1 will receive the input traffic from port 3/1 as well as port 4/1 and mirror port 1/2 will receive input traffic from port 4/1 as well as port 3/1 even if they are not explicitly configured to do so. The outbound traffic from port 3/1 is mirrored to port 1/1 only, as configured and the outbound traffic from port 4/1 is mirrored to port 1/2 only as configured.

This example also configures one mirror port 2/1 on another module, to which inbound traffic from port 3/1 is mirrored. Because only one mirror port is configured on this module, the traffic is mirrored as configured.

If input monitoring is enabled on two ports controlled by the same packet processor, then the input traffic on these two ports will be mirrored to all the ports configured as mirror ports for these two monitored ports. This restriction does not apply to outbound monitoring.

```
BigIron RX(config)# mirror-port ethernet 1/1
BigIron RX(config)# mirror-port ethernet 2/1
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e1000-3/1)# monitor ethernet 1/1 both
BigIron RX(config-if-e1000-3/1)# interface ethernet 3/2
BigIron RX(config-if-e1000-3/2)# monitor ethernet 2/1 both
```

The above example configures two mirror ports 1/1 and 2/1 on different modules. Port 3/1 uses port 1/1 for inbound and outbound mirroring. Port 3/2 uses port 2/1 for inbound and outbound mirroring. If 3/1 and 3/2 are controlled by the same packet processor, inbound traffic from 3/1 will be mirrored to 1/1 as well as 2/1 and similarly, inbound traffic from 3/2 will be mirrored to 2/1 as well as 1/1. The outbound traffic on 3/1 and 3/2 are mirrored according to the configuration.

Monitoring an individual trunk port

By default, when you monitor the primary port in a trunk group, aggregated traffic for all the ports in the trunk group is copied to the mirror port. You can configure the device to monitor individual ports in a trunk group. You can monitor the primary port or a secondary port individually.

NOTE
You can use only one mirror port for each monitored trunk port.

To monitor traffic on an individual port in a trunk group, enter commands such as the following.

```
BigIron RX(config)# mirror ethernet 2/1
BigIron RX(config)# trunk switch ethernet 4/1 to 4/8
BigIron RX(config-trunk-4/1-4/8)# config-trunk-ind
BigIron RX(config-trunk-4/1-4/8)# monitor ethe-port-monitored 4/5 ethernet 2/1 in
```

Syntax: [no] config-trunk-ind

Syntax: [no] monitor ethe-port-monitored <portnum> | named-port-monitored <portname> ethernet <slot>/<portnum> in | out | both

The **config-trunk-ind** command enables configuration of individual ports in the trunk group. You need to enter the **config-trunk-ind** command only once in a trunk group. After you enter the command, all applicable port configuration commands apply to individual ports only.

NOTE
If you enter **no config-trunk-ind**, all port configuration commands are removed from the individual ports and the configuration of the primary port is applied to all the ports. Also, once you enter the **no config-trunk-ind** command, the **enable**, **disable**, and **monitor** commands are valid only on the primary port and apply to the entire trunk group.

The **monitor ethe-port-monitored** command in this example enables monitoring of the inbound traffic on port 4/5:

- The **ethe-port-monitored** <portnum> | **named-port-monitored** <portname> parameter specifies the trunk port you want to monitor. Use **ethe-port-monitored** <portnum> to specify a port number. Use **named-port-monitored** <portname> to specify a trunk port name.
- The **ethernet** <slot>/<portnum> parameter specifies the port to which the traffic analyzer is attached.
- The **in | out | both** parameter specifies the traffic direction to be monitored.

Mirror ports for Policy-Based Routing (PBR) traffic

You can mirror traffic on ports that have policy-based routing (PBR) enabled. This feature is useful for monitoring traffic, debugging, and enabling application-specific mirroring.

The PBR mirror interface feature allows continued hardware forwarding and, at the same time, enables you to determine exactly which traffic flows get routed using the policies defined by PBR.

The following section provides a general overview of hardware-based PBR.

About hardware-based PBR

Hardware-based Policy-Based Routing (PBR) routes traffic in hardware based on policies you define. A PBR policy specifies the next hop for traffic that matches the policy. A PBR policy also can use an ACL to perform QoS mapping and marking for traffic that matches the 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. You also can map and mark the traffic's QoS information using the QoS options of the ACLs.

Configuring mirror ports for PBR traffic

When you configure a physical or virtual port to act as a mirror port for PBR traffic, outgoing packets that match the permit Access Control List (ACL) clause in the route map are copied to the mirror ports that you specify. You can specify up to four mirror ports for each PBR route map instance.

For example, to capture all traffic forwarded to an SSL port and mirror it to port 5, enter commands such as the following.

```
BigIron RX(config)# route-map ssl-pbr-map permit 1
BigIron RX(config-routemap ssl-pbr-map)# match ip address 100
BigIron RX(config-routemap ssl-pbr-map)# set mirror-interface 5
BigIron RX(config-routemap ssl-pbr-map)# set next-hop 10.10.10.1
BigIron RX(config-routemap ssl-pbr-map)# exit
BigIron RX(config)# interface e 5
BigIron RX(config-if-e10000-5)# port-name mirror-port
BigIron RX(config-if-mirror-port)# interface e 10
BigIron RX(config-if-mirror-port-10)# ip policy route-map ssl-pbr-map
BigIron RX(config-if-mirror-port-10)# exit
BigIron RX(config-if-e10000-)#exit
BigIron RX(config)#access-list 100 permit tcp any any eq ssl
```

The above commands complete the following configuration tasks.

1. Configures an entry in the PBR route map named “ssl-pbr-map” . The match statement matches on IP information in ACL 100. The **set mirror-interface** statement specifies interface e 5 as the mirror port for matched ACL permit clauses. The **set next-hop** statement sets the IP address of the route’s next hop router to 10.10.10.1.
2. Identifies interface e 5 as a mirror port by assigning the name “*mirror-port*”.
3. Enables PBR and applies the route map *“ssl-pbr-map”* on interface e 10.
4. Creates an extended ACL (100) that permits all TCP traffic destined for an for an SSL port.

Syntax: set mirror-interface <slot number>/<port number>

The <*slot number*> parameter specifies the port number on a device.

The <*port number*> parameter specifies the mirror port number.

You can specify up to 4 mirror ports for each PBR route map instance. To do so, enter the **set mirror interface** command for each mirror 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.

```
BigIron RX# show monitor config
Monitored Port 3/1
    Input traffic mirrored to: 1/1 2/1
    Output traffic mirrored to: 1/1
Monitored Port 4/1
    Input traffic mirrored to: 1/2
    Output traffic mirrored to: 1/2
```

Syntax: show monitor config

This output does not display the input traffic mirrored to mirror port 1/2 from port 3/1 and mirrored to mirror port 1/1 from port 4/1 because the mirroring of this traffic is not explicitly configured.

To display the actual traffic mirrored to each mirror port, enter the following command at any level of the CLI.

```
BigIron RX# show monitor actual
Monitored Port 3/1
  Input traffic mirrored to: 1/1(configured) 1/2 2/1(configured)
  Output traffic mirrored to: 1/1
Monitored Port 4/1
  Input traffic mirrored to: 1/2(configured) 1/1
  Output traffic mirrored to: 1/2
```

Syntax: show monitor actual

This output displays the input traffic mirrored to mirror port 1/2 from port 3/1 and mirrored to mirror port 1/1 from port 4/1, which are not explicitly configured.

Enabling WAN PHY mode support

A 10 Gigabit Ethernet port can be configured to use SONET/SDH framing for Layer 1 transport across a WAN transport backbone by configuring the port in WAN PHY mode. The default is for the port to operate in LAN PHY mode.

To enable a 10 GB Ethernet port to support WAN PHY mode, use the following command.

```
BigIron RX#(config-if-e10000-6/3)# phy-mode wan
```

Syntax: [no] phy-mode wan

To change the PHY mode for a port back to the default of LAN PHY mode, use the **no** condition before the command.

Chapter

7

Configuring IP

In this chapter

Overview of configuring IP

The Internet Protocol (IP) is enabled by default. This chapter describes how to configure IP parameters on the device.

The IP packet flow

Figure 5 Shows how an IP packet moves through a device.

FIGURE 5 IP Packet flow through a device

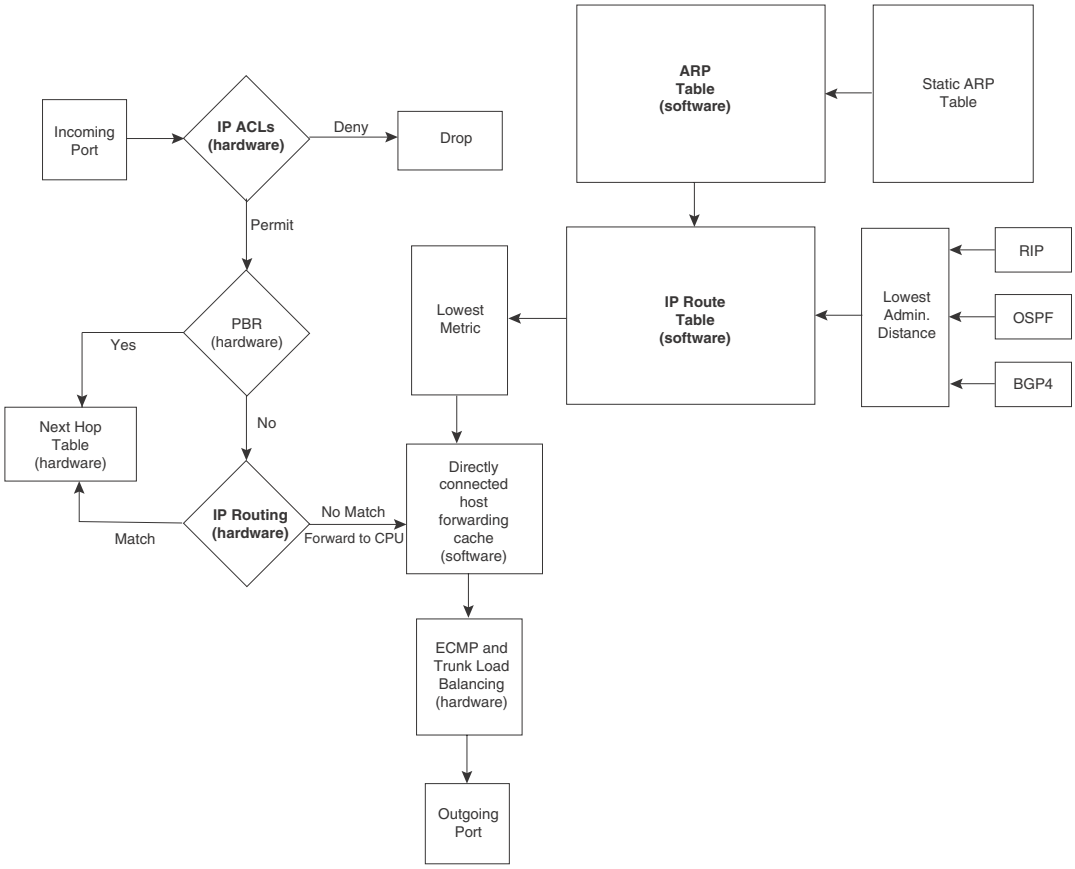


Figure 5 Shows the following packet flow.

1. When the device receives an IP packet, the device checks for IP ACL filters on the receiving interface. If a deny filter on the interface denies the packet, the device discards the packet and performs no further processing. If logging is enabled for the filter, then the device generates a Syslog entry and SNMP trap message.
2. If the packet is not denied, the device 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 device 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 trunk 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.

ARP cache table

The Address Resolution Protocol (ARP) is supported on the device. Refer to "IP fragmentation protection" on page 178.

The ARP cache contains entries that map IP addresses to MAC addresses. Generally, the entries are for devices that are directly attached to the device.

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 device learns a device's MAC address from an ARP request or ARP reply from the device.

The software can learn an entry when the device receives an ARP request from another IP forwarding device or an ARP reply. Here is an example of a 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 device 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 device.

The software places an entry from the static ARP table into the ARP cache when the entry's interface comes up.

Here is an example of a 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 217
- "Displaying the static ARP table" on page 218

To configure other ARP parameters, refer to "IP fragmentation protection" on page 178.

To increase the size of the ARP cache and static ARP table, see the following:

- For dynamic entries, refer to "Displaying and modifying system parameter default settings" on page 127. The ip-arp parameter controls the ARP cache size.
- For static entries, refer to "Changing the maximum number of entries the static ARP table can hold" on page 184. The ip-static-arp parameter controls the static ARP table 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 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.

Here is an example of an entry in the 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 220.

To configure a static IP route, refer to "Configuring static routes" on page 191.

To clear a route from the IP route table, refer to "Clearing IP routes" on page 223.

To increase the size of the IP route table for learned and static routes, refer to "Displaying and modifying system parameter default settings" on page 127:

- For learned routes, modify the ip-route parameter.
- For static routes, modify the ip-static-route parameter.

IP forwarding cache

The device 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.

Here is an example of an entry in the 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 device 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 219.

Basic IP parameters and defaults

IP is enabled by default. The following protocols are disabled by default:

- Route exchange protocols (RIP, OSPF, 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 41 lists the IP global parameters for the device, their default values, and where to find configuration information.

TABLE 41 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 154
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 174
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 173
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 179
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 180
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 42 on page 152.	Ten minutes	page 182
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 182
Static ARP entries	An ARP entry you place in the static ARP table. Static entries do not age out.	2048	page 183

TABLE 41 IP global parameters (Continued)

Parameter	Description	Default	See page...
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 186
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 42 on page 152.	Disabled	page 187
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 188
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 187
Internet Control Message Protocol (ICMP) messages	The device 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 42 on page 152.	Enabled	page 188 page 190
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 42 on page 152.	Disabled	page 206
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 212
Maximum Frame Size	You can set a maximum frame size of IP packets that are forwarded on all ports of a PPCR.		page 172
Domain name for Domain Name Server (DNS) resolver	A domain name (example: foundry.router.com) 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 166
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 166

TABLE 41 IP global parameters (Continued)

Parameter	Description	Default	See page...
IP load sharing	A *Brocade* 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 201
Maximum IP load sharing paths	The maximum number of equal-cost paths across which the device is allowed to distribute traffic.	Four	page 201
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 669 page 706 page 764
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 200
Static route	An IP route you place in the IP route table.	No entries	page 191
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 175

IP interface parameters

Table 42 lists the interface-level IP parameters for the device, their default values, and where to find configuration information.

TABLE 42 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 device has separate IP addresses on individual interfaces.	None configured[a]	page 154
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 171

TABLE 42 IP interface parameters (Continued)

Parameter	Description	Default	See page...
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 173
ARP age	Locally overrides the global setting. Refer to Table 41 on page 150.	Ten minutes	page 182
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	page 666
Directed broadcast forwarding	Locally overrides the global setting. Refer to Table 41 on page 150.	Disabled	page 187
ICMP Router Discovery Protocol (IRDP)	Locally overrides the global IRDP settings. Refer to Table 41 on page 150.	Disabled	page 207
ICMP Redirect messages	Locally overrides the global setting. Refer to Table 41 on page 150.	Enabled	page 190
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 212
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. See the next row.	The router helps forward broadcasts for the following UDP application protocols: • bootps • dns • netbios-dgm • netbios-ns • tacacs • tftp • time	page 209
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 210

a. Some devices have a factory default, such as 209.157.22.154, used for troubleshooting during installation. For the device, the address is on module 1 port 1 (or 1/1).

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 device 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.

Also, the CAM can hold up to 256,000 IP address entries.

NOTE
Once 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, once 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 will not be bridged but dropped.

The device 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. Refer to “Changing the network mask display to prefix format” on page 156.

Assigning an IP address to an Ethernet port

To assign an IP address to port 1/1, enter the following commands.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(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.

```
BigIron RX(config-if-e10000-1/1)# ip address 192.45.6.1/24
```

Syntax: 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 device 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 device and other devices.

You can configure up to eight loopback interfaces on a device.

You can add up to 24 IP addresses to each loopback interface.

NOTE
If you configure the device 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 device. Refer to "Adding a loopback interface" on page 789 in the BGP4 chapter.

To add a loopback interface, enter commands such as those shown in the following example.

```
BigIron RX(config-bgp-router)# exit
BigIron RX(config)# int loopback 1
BigIron RX(config-lbif-1)# ip address 10.0.0.1/24
```

Syntax: interface loopback <num>

For the syntax of the IP address, refer to "Assigning an IP address to an Ethernet port" on page 154.

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 device.

NOTE
Other sections in this chapter that describe how to configure interface parameters also apply to virtual interfaces.

NOTE
The device 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.

```
BigIron RX(config)# vlan 2 name IP-Subnet_1.1.2.0/24
BigIron RX(config-vlan-2)# untag e1/1 to 1/4
BigIron RX(config-vlan-2)# router-interface ve1
BigIron RX(config-vlan-2)# interface ve1
BigIron RX(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: router-interface ve <num>

Syntax: 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 154.

Deleting an IP address

To delete an IP address, enter a command such as the following.

```
BigIron RX(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.

```
BigIron RX(config-if-e1000-1/1)# no ip address *
```

Syntax: no ip address <ip-addr>

Changing the network mask display to prefix format

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). 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 ACL entries regardless of whether the software is configured to display the masks in CIDR format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can change the format to the CIDR prefix format (example: /18) by entering the following CLI command.

```
BigIron RX(config)# ip show-subnet-length
```

Syntax: [no] ip show-subnet-length

Configuring the default gateway

To manage a device using Telnet or Secure Shell (SSH) CLI connections or the Web management interface, you must configure an IP address for the device.

To configure a default gateway, first define an IP address using the following CLI command.

```
BigIron RX(config)# ip address 192.45.6.110 255.255.255.0
```

Syntax: ip address <ip-addr> <ip-mask>

or

Syntax: ip address <ip-addr>/<mask-bits>

GRE IP tunnel

The BigIron RX allows the tunneling of packets of the following protocols over an IP network using the Generic Router Encapsulation (GRE) mechanism as described in RFC 2784:

- OSPF
- BGP
- IS-IS point-to-point

Using this feature, packets of these protocols can be encapsulated inside a transport protocol packet at a tunnel source and delivered to a tunnel destination where it is unpacked and made available for delivery. Figure 6 describes the GRE header format.

FIGURE 6 GRE header format

1 bit Checksum	12 bits Reserved0	3 bits Ver	16 bits Protocol Type	16 bits Checksum (optional)	16 bits Reserved (optional)

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 zero in transmitted packets. If bits 11:7 of the field are non-zero, then a receiver must discard the packet unless RFC 1701 is implemented. This field is assumed to be zero in this version.

Ver – This field must be set to zero. This field is assumed to be zero in this version.

GRE MTU configuration considerations

The default value of IP GRE tunnel MTU is 1476 bytes. The MTU of the GRE tunnel is compared with the outgoing packet before the encapsulation is done. After the encapsulation, the packet size increases by 24 bytes. If a user wants to change the GRE tunnel MTU, the MTU should be at least 24 bytes less than the IP MTU of the outgoing interface. Otherwise, the size of the encapsulated packet will exceed the IP MTU of the outgoing interface. In that case, the packet is dropped if the DF (Do-Not-Fragment) bit is set in the original IP packet, otherwise, the packet is sent to CPU for fragmentation.

NOTE
The encapsulated packets sent on a GRE tunnel have the DF bit set. Setting a GRE tunnel MTU to be greater than 1476 will cause the encapsulated packet to be greater than 1500 bytes. This may cause the transit routers to drop the encapsulated packet if that transit router's IP MTU is 1500 bytes (a typical default MTU value) since transit routers can not fragment a GRE packet.

Configuring a GRE IP tunnel

To configure a GRE IP Tunnel, the following parameters must be configured:

- Tunnel interface
- Source Address for the Tunnel
- Destination address for the Tunnel
- GRE Encapsulation
- Loopback address for the Tunnel (required for de-encapsulation)
- IP address for the Tunnel

NOTE
Sustained rates of small packet sizes may affect the ability of a 10 gigabit Ethernet port to maintain line rate GRE encapsulation and de-encapsulation performance.

Configuring a tunnel interface

To configure a tunnel interface, use a the following command.

```
BigIron RX(config)# interface tunnel 1
BigIron RX(config-tnif-1)
```

Syntax: interface tunnel <tunnel-number>

The <*tunnel-number*> variable is numerical value that identifies the tunnel being configured.

Configuring a source address for a tunnel interface

To configure a source address for a specific tunnel interface, enter the following command.

```
BigIron RX(config)# interface tunnel 1
BigIron RX(config-tnif-1)tunnel source 35.0.8.108
```

Syntax: tunnel source <*ip-address*>

The <*ip-address*> variable is source IP address being configured for the specified tunnel.

Configuring a destination address for a tunnel interface

To configure a destination address for a specific tunnel interface, enter the following command.

```
BigIron RX(config)# interface tunnel 1
BigIron RX(config-tnif-1)tunnel destination 131.108.5.2
```

Syntax: tunnel destination <*ip-address*>

The <*ip-address*> variable is destination IP address being configured for the specified tunnel.

NOTE
Ensure a route to the tunnel destination exist on the tunnel source device. Create a static route if needed.

Configuring a tunnel interface for GRE encapsulation

To configure a specified tunnel interface for GRE encapsulation, enter the following command.

```
BigIron RX(config)# interface tunnel 1
BigIron RX(config-tnif-1)tunnel mode gre ip
```

Syntax: 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 a loopback port for a tunnel interface

On the device, a loopback port is required for de-encapsulating a packet exiting the tunnel.

Fiber-optic components must be present on the interface module for the loopback port to work. Therefore, consider the following configuration rules for a loopback port:

- 1-gigabit copper ports should not be configured as loopback ports.
- 1-gigabit and 10-gigabit fiber ports can be configured as loopback port:
  - 1-gigabit fiber ports require a fiber cable to be connected to itself for loopback to work.
  - 10-gigabit fiber ports do not require a cable.

To configure a loopback port for a specified tunnel interface, enter the following commands.

```
BigIron RX(config)# interface tunnel 1
BigIron RX(config-tnif-1)tunnel loopback 3/1
```

Syntax: tunnel loopback <port-number>

The <*port-number*> variable is the port number assigned to be the loopback port for the specified tunnel interface. A loopback port is required to perform termination and forwarding in hardware. If a loopback port is not configured, tunnel termination is performed by the CPU. When a port is used as a loopback port for a tunnel, it should not be used for any other purpose.

NOTE
The tunnel loopback port is one of the router's physical ports. It is defined so the GRE packet processing Is done on by the port's LP CPU instead of the MP's CPU. You can use a 10 GBE port without a loopback connector but the optical transceiver module MUST be installed. You can use a 1 GBE fiber port, but a physical loopback connector is required. Copper ports are not supported.

Configuring an IP address for a tunnel interface

To configure an IP address for a specified tunnel interface, enter the following command.

```
BigIron RX(config)# interface tunnel 1
BigIron RX(config-tnif-1)ip address 10.10.3.1/24
```

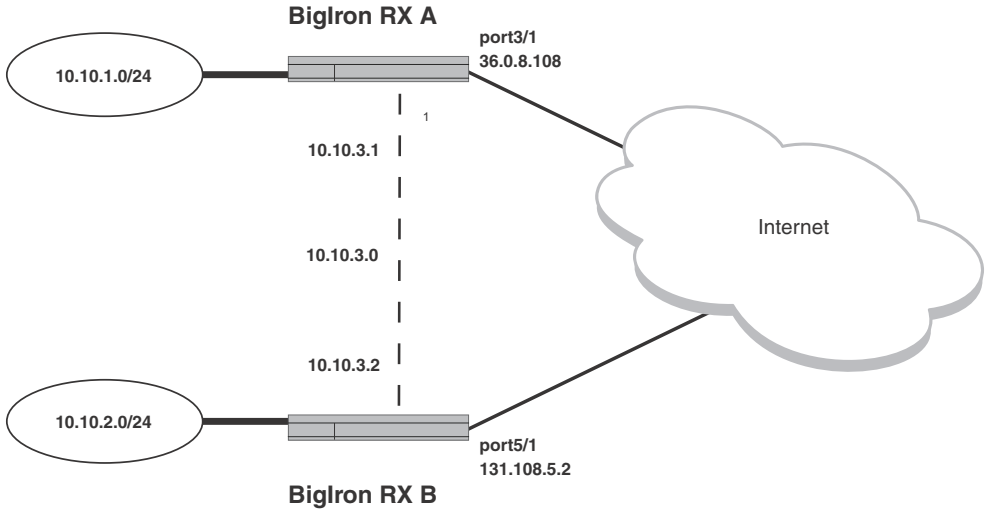
Syntax: ip address <*ip-address*>

The <*ip-address*> variable is the IP address being configured for the specified tunnel interface.

Example of a GRE IP tunnel configuration

In this example, a GRE IP Tunnel is configured between the device A switch and the device B switch. 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 7 GRE IP tunnel configuration example



Configuration example for BigIron RX A

```
BigIron RX (config)# interface ethernet 3/1
BigIron Rx (config-if-e1000-3/1)# ip address 36.0.8.108/24
BigIron RX (config)# exit
BigIron RX (config)# interface tunnel 1
BigIron RX(config-tnif-1)# tunnel loopback 4/1
BigIron RX(config-tnif-1)# tunnel source 36.0.8.108
BigIron RX(config-tnif-1)# tunnel destination 131.108.5.2
BigIron RX(config-tnif-1)# tunnel mode gre ip
BigIron RX(config-tnif-1)# ip address 10.10.3.1/24
BigIron RX(config-tnif-1)# exit
BigIron RX (config)# ip route 131.108.5.0/24 36.0.8.1
BigIron RX(config)# ip route 10.10.2.0/24 tunnel 1
```

Configuration example for BigIron RX B

```
BigIron RX(config)# interface ethernet 5/1
BigIron RX(config--if-e1000-5/1)# ip address 131.108.5.2/24
BigIron RX (config)# exit
BigIron RX (config)# interface tunnel 1
BigIron RX(config-tnif-1)# tunnel loopback 1/1
BigIron RX(config-tnif-1)# tunnel source 131.108.5.2
BigIron RX(config-tnif-1)# tunnel destination 36.0.8.108
BigIron RX(config-tnif-1)# tunnel mode gre ip
BigIron RX(config-tnif-1)# ip address 10.10.3.2/24
BigIron RX(config-tnif-1)# exit
BigIron RX(config)# ip route 36.0.8.0/24 131.108.5.1
BigIron RX(config)# ip route 10.10.1.0/24 tunnel 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.

```
BigIron RX# show ip interface tunnel 1
   Interface       IP-Address      OK?  Method    Status    Protocol  VRF
   Tunnel 1        10.10.3.1      YES  NVRAM     up        up        default
```

Syntax: show ip interface tunnel <tunnel-no>

This display shows the following information.

TABLE 43 CLI display of interface IP configuration information

This field...	Displays...
Interface	The tunnel and tunnel number.
IP-Address	The IP address of the tunnel interface.
OK?	Whether the IP address has been configured on the tunnel 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”.
VRF	The name of the Virtual Routing instance that the tunnel is configured in.

The **show ip route** command displays routes that are pointing to a GRE tunnel as shown in the following.

```
BigIron RX# show ip route
Total number of IP routes: 9
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
    Destination     Gateway       Port      Cost     Type
1   2.2.2.1/32      DIRECT        loopback1  0/0      D
2   10.10.1.0/24    110.110.2.12  tunnel 1   1/1      S
3   20.2.1.0/24     DIRECT        eth5/11    0/0      D
4   45.4.1.0/24     80.8.1.2      tunnel 2   0/0      D
5   63.148.1.0/24   DIRECT        eth 2/11   0/0      D
6   70.7.1.0/24     DIRECT        eth 2/14   0/0      D
7   80.8.1.0/24     70.7.1.1      eth 2/14   1/1      S
8   110.110.2.0/24  63.148.1.1    eth 2/11   1/1      S
9   189.100.1.0/24  110.110.2.12  tunnel 1   0/0      D
```

The **show interface tunnel** command displays the status and configuration information for a tunnel interface as shown in the following.

```
BigIron RX# show interface tunnel 1
    Tunnel1 is up, line protocol is up
    Hardware is Tunnel
    Tunnel source  63.148.1.2
    Tunnel destination is 110.110.2.12
```

```
Tunnel mode gre ip
Tunnel loopback is 1/3
No port name
MTU 1476 Bytes
```

Syntax: show interface tunnel <*number*>

The <*number*> parameter indicates the tunnel interface number for which you want to display information.

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.

Brocade supports the following IPv6 over IPv4 tunneling in hardware mechanisms:

- Manually configured tunnels

In general, a manually configured tunnel establishes a permanent link between routers in IPv6 domains. A manually configured tunnel has explicitly configured IPv4 addresses for the tunnel source and destination.

This tunneling mechanism requires 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.

Configuring a manual IPv6 tunnel

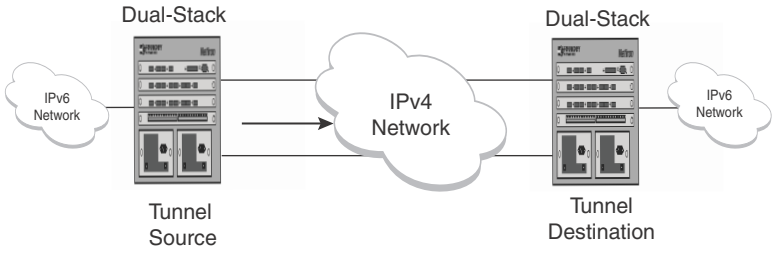
You can use a manually configured tunnel to connect two isolated IPv6 domains. You should deploy this point-to-point tunnel mechanism if you need a permanent and stable connection.

Configuration notes

- 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 we need to have exactly opposite source or 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
- Both static and dynamic IPv6 routing protocols on top of the tunnel are supported
- The tunnel source can be ip address or interface name
- Manual tunnels provide static point-point connectivty

NOTE
IPV6 over IPV4 tunnel will not work when used with transperant VLAN flooding mode .

FIGURE 8 Manually configured tunnel



To configure a manual IPv6 tunnel, enter commands such as the following on a Layer 3 Switch running both IPv4 and IPv6 protocol stacks on each end of the tunnel.

```
BigIron RX(config)# interface tunnel 1
BigIron RX(config-tnif-1)#tunnel source ethernet 3/1
BigIron RX(config-tnif-1)#tunnel destination 198.162.100.1
BigIron RX(config-tnif-1)#tunnel mode ipv6ip
BigIron RX(config-tnif-1)#ipv6 address 2001:b78:384d:34::/64 eui-64
```

This example creates tunnel interface 1 and assigns a global IPv6 address with an automatically computed EUI-64 interface ID to it. The IPv4 address assigned to Ethernet interface 3/1 is used as the tunnel source, while the IPv4 address 192.168.100.1 is configured as the tunnel destination. Finally, the tunnel mode is specified as a manual IPv6 tunnel.

Syntax: interface tunnel <number>

For the <number> parameter, specify a value between 1 – 32.

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 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: tunnel source <ipv4-address> | ethernet <port> | loopback <number> | ve <number>

You must specify the <*ipv4-address*> parameter using 8-bit values in dotted decimal notation.

The **ethernet | loopback | ve** parameter specifies an interface as the tunnel source. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, VE, or interface, also specify the loopback, VE, or number, respectively.

Syntax: tunnel destination <*ipv4-address*>

You must specify the <*ipv4-address*> parameter using 8-bit values in dotted decimal notation.

Syntax: tunnel mode ipv6ip

Clearing IPv6 tunnel statistics

You can clear all IPv6 tunnel statistics (reset all fields to zero) or statistics for a specified tunnel interface.

For example, to clear statistics for tunnel 1, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
BigIron RX# clear ipv6 tunnel 1
```

Syntax: clear ipv6 tunnel <*number*>

The <*number*> parameter specifies the tunnel number.

Displaying IPv6 tunnel information

To display a summary of tunnel information, enter the following command at any level of the CLI.

```
BigIron RX# show ipv6 tunnel
IP6 Tunnels
  Tunnel  Mode         Packet Received  Packet Sent
  1       configured   0                0
  2       configured   0                22419
```

Syntax: show ipv6 tunnel

This display shows the following information.

TABLE 44 IPv6 tunnel information

This field...	Displays...
Tunnel	The tunnel interface number.
Mode	The tunnel mode. Possible modes include the following: • configured – Indicates a manually configured tunnel. • 6to4 – Indicates an automatic 6to4 tunnel. • auto – Indicates an automatic IPv4-compatible tunnel.
Packet Received	The number of packets received by a tunnel interface.
Packet Sent	The number of packets sent by a tunnel interface.

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.

```
BigIron RX# 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 45 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.
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.

```
BigIron RX#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.

```
BigIron RX#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
```

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 device and thereby 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 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.

```
BigIron RX# ping nyc01
BigIron RX# 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 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, enter the following commands.

```
BigIron RX(config)# ip dns domain-name newyork.com
BigIron RX(config)# ip dns server-address 209.157.22.199 205.96.7.15 208.95.7.25
201.98.7.15
```

Syntax: ip dns domain-name <*name*>

Syntax: ip dns server-address <*ip-addr*> [<*ip-addr*>] [<*ip-addr*>] [<*ip-addr*>]

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.

Defining a domain list

If you want to use more than one domain name to resolve host names, you can create a list of domain names. For example, enter the commands such as the following.

```
BigIron RX(config)# ip dns domain-list company.com
BigIron RX(config)# ip dns domain-list ds.company.com
BigIron RX(config)# ip dns domain-list hw_company.com
BigIron RX(config)# ip dns domain-list qa_company.com
BigIron RX(config)#
```

The domain names are tried in the order you enter them.

Syntax: [no] ip dns domain-list <*domain-name*> sequence-number <*sequence-number*>

The <*domain-name*> parameter specifies the domain name to be added to the list.

The <*sequence-number*> parameter specifies a sequence number that is generated internally in steps of 10 starting with sequence number 5. The entries are tried in order of sequence number of entries.

Use the **no** form of the command to remove a domain name from the domain-list.

Displaying the domain name list

To determine what domain names have been configured in the domain list, enter the following command.

```
BigIron RX(config)#show ip dns domain-list
Total number of entries : 3
Primary Domain Name:
Domain Name List:
seq:4 eng.company.co
seq:5 facilities.company.com
seq:12.   support.company.com
```

Syntax: show ip dns domain-list

Verifying domain name or IP address

You can use the **ip domain-lookup** command to verify the host name for an IP address or the IP address for a host name. For example, if you have an IP address and you want to find out what host name it resolves to, enter the following command.

```
BigIron RX#ip domain-lookup 66.151.144.5
Host Flag TTL/min Type Address
border2.pc0-0-bbnet1.sje.pnap.net (TMP,OK) 720 IP 66.151.144.5
```

You can also enter the following.

```
BigIron RX#ip domain-lookup border2
Host Flag TTL/min Type Address
border2.pc0-0-bbnet1.sje.pnap.net (TMP,OK) 720 IP 66.151.144.5
```

Syntax: ip domain-loopkup <*ip-address*> | <*host-name*>

<*ip-address*> Enter an IP address to obtain the host name.

<*host-name*> Enter the host name to obtain the IP address.

The complete, qualified host name, along with its IP address and TTL value are displayed.

Adding host names to the DNS cache table

Dynamic cache entries

The entries in a DNS cache table are used to resolve host names to IP addresses. When a client initiates a DNS query, the *Brocade* device checks the DNS cache table to see if the host name can be resolved to any of the entries. If it a match is found, the query is resolved. If a match is not found, the DNS resolver sends the query to the DNS servers. If the name is resolved, the complete, qualified host name and its IP address is added to the DNS cache table and the hosts' IP address is returned to the client.

Static cache entries

You can manually add entries to the DNS cache table if you know a host's complete, qualified name and its IP address. To add host names and their IP addresses to the DNS cache table, enter commands such as the following.

```
BigIron RX(config)#ip dns cache-entry www.foundrynet.com 63.236.63.244 720
```

Syntax: [no] ip dns cache-entry <*host-name*> <*ip-address*>

<*host-name*> Complete, qualified name . For example, enter www.company.com or host.company.com.

<*ip-address*> Enter the IP address of the host. This must be the correct IP address for the host.

Use the **no** form of the command to manually remove an entry from the DNS cache table; however, you must enter the entire entry to delete the entry.

Example

```
BigIron RX(config)#no ip dns cache-entry www.foundrynet.com 63.236.63.244
```

Clearing the DNS cache table

To clear the entire DNS cache table, enter the following command.

```
BigIron RX#clear ip dns cache-table
```

To clear a specific entry in DNS cache table, enter the following command.

```
BigIron RX# clear ip dns cache-table www.foundrynet.com
```

OR

```
BigIron RX# clear ip dns cache-table 63.236.63.244
```

Syntax: clear ip dns cache-table [ip-address | host-name]

<*host-name*> Complete, qualified name . For example, enter www.company.com or host.company.com.

<*ip-address*> Enter the IP address of the host. This must be the correct IP address for the host.

Displaying the DNS cache table

To display what hosts are currently in the DNS cache table, enter the following command.

```
BigIron RX(config)#show ip dns cache-table
Host                            Flag          Address
border2.pc0-0-bbnet1.sje.pnap.net (TMP,OK)     66.151.144.5
sl-internap-109-0.sprintlink.net  (TMP,OK)     144.223.242.86
sl-st21-sj-13-0.sprintlink.net    (TMP,OK)     144.232.20.59
mail.company.com                     (STA,OK)      64.236.22.148
```

To display the individual entries in the cache-table, enter a command such as the following.

```
BigIron RX(config)#show ip dns cache-table border2
Host Flag TTL/min Address
border2.pc0-0-bbnet1.sje.pnap.net (TMP,OK) 720 66.151.144.5
```

OR

```
BigIron RX(config)#show ip dns cache-table 66.151.144.5
Host Flag TTL/min Address
border2.pc0-0-bbnet1.sje.pnap.net (TMP,OK) 720 66.151.144.5
```

TABLE 46 The show ip dns cache-table output

This field...	Displays...
Host	The complete, qualified domain name of the host.
Flag	Indicates if the entry is dynamic or static and if the information for the domain is up to date: • TMP – Entry is dynamic • STA – Entry is static • OK – Information for the entry is up to date • EX – The entry is expired and would not be used. Such an entry would be deleted from the cache table at next cache poll refresh.
TTL/min	If the entry is dynamic (TMP) this value shows how long the entry remains in the DNS cache table. If the entry is static (STA), it remains in the DNS cache table and never changes until it is manually removed or the DNS cache table is cleared.
Address	The IP address of the entry.

Syntax: show ip dns cache-table [host-name | ip-address]

<*host-name*> Complete, qualified name . For example, enter www.company.com or host.company.com.

<*ip-address*> Enter the IP address of the host. This must be the correct IP address for the host.

Defining the polling interval

The polling interval determines how often the *Brocade* device checks the status of the entries in the DNS cache table to determine if the information for that host has changed. If the TTL value of the cache entry is expired the entry is removed from the cache-table.

To define a polling interval, enter the following command.

```
BigIron RX(config)#ip dns poll-interval 7
```

Syntax: ip dns poll-interval <minutes>

Enter the polling interval in minutes. The default is 1 minutes.

Displaying the polling interval

To display the current polling interval configured for the device, enter the following command.

```
BigIron RX(config)#show ip dns poll-time-interval
Current DNS polling interval is 7 minutes
```

Syntax: show ip dns poll-time-interval

Displaying the server list

To display the current DNS server list configured for the device, enter the following command.

```
BigIron RX#show ip dns server-list
Total number of DNS Servers configured: 2
Server List:
10.51.17.30
10.51.17.29
```

Syntax: show ip dns server-list

Debugging the DNS feature

To debug the DNS feature enter the following command.

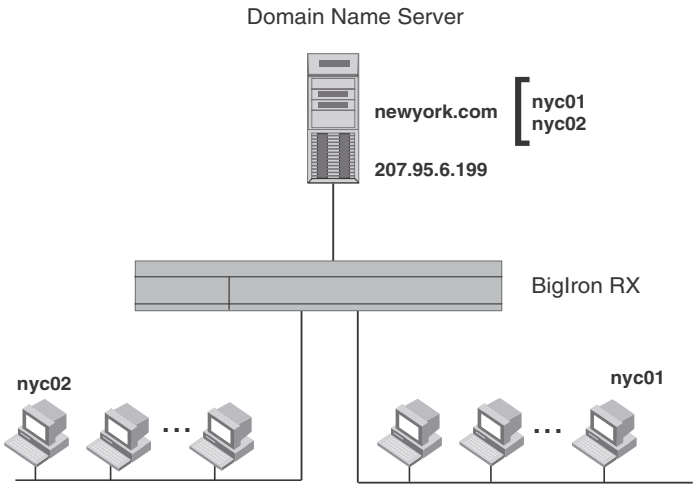
```
BigIron RX#debug ip dns
IP: dns debugging is on
```

Syntax: debug ip dns

Using a DNS name to initiate a trace route

Suppose you want to trace the route from a device to a remote server identified as NYC02 on domain newyork.com.

FIGURE 9 Querying a host on the newyork.com domain



Because the newyork.com domain is already defined on the device, you need to enter only the host name, NYC02, as noted below.

```
BigIron RX# traceroute nyc02
```

Syntax: 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.

Configuring packet parameters

You can configure the following packet parameters to control how the device sends IP packets to other devices on an Ethernet network. The device always places IP packets into Ethernet packets to forward them on an Ethernet port:

- **Encapsulation type** – The format for the Layer 2 packets within which the device sends IP packets.
- **Maximum Frame Size** – The maximum frame size that applies to all ports on a packet processor (PPCR).
- **IP Maximum Transmission Unit (MTU)** – The maximum length of IP packet that a Layer 2 packet can contain. IP packets that are longer than the IP MTU are fragmented and sent in multiple Layer 2 packets. You can change the IP MTU globally or on a port:
  - **Global IP MTU** – The default IP MTU value depends on the encapsulation type on a port and is 1500 bytes for Ethernet II encapsulation and 1492 bytes for SNAP encapsulation.
  - **Port IP MTU** – A port's default IP MTU depends on the encapsulation type enabled on the port.

Changing the encapsulation type

The device 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 device 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 device.
- 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 device 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 device port must use the same encapsulation type.

To change the IP encapsulation type on interface 1/5 to Ethernet SNAP, enter the following commands.

```
BigIron RX(config)# int e 1/5
BigIron RX(config-if-e1000-1/5)# ip encapsulation snap
```

Syntax: ip encapsulation snap | ethernet-2

Setting maximum frame size per PPCR

You can set a maximum frame size of IP packets that are forwarded on all ports of a PPCR. You can set a maximum frame size globally and per interface.

Globally setting the maximum frame size

To set a maximum frame size that applies to the device, enter a command such as the following.

```
BigIron RX(config)# default-max-frame-size 2000
BigIron RX(config)# write memory
BigIron RX(config)# reload
```

Syntax: default-max-frame-size <*bytes*>

Enter 64 – 9212 for <*bytes*>. The default is 1518 bytes.

Setting a maximum frame size per interface

When you set a maximum frame size on an interface, that size applies to all ports in a PPCR. Table 47 shows the ports of each Interface module.

TABLE 47 Available ports per PPCR

Module type	Number of packet processors (PPCR)	Ports in a PPCR			
		PPC1	PPCR2	PPCR3	PPCR4
24 x 1G	2	1 - 12	13 - 24	N/A	N/A

To set a maximum frame size for all the ports attached to a PPCR, enter a command such as the following at the Interface Configuration level.

```
BigIron RX(config)#interface ethernet 6/4
BigIron RX(config-if-e1000-6/4)#max-frame-size 1500 bytes
BigIron RX(config-if-e1000-6/4)#write memory
BigIron RX(config-if-e1000-6/4)#exit
BigIron RX(config)#reload
```

In this example the maximum frame size is applied to port 4 of a 24 x 1G Ethernet Interface module. That means that this maximum will apply to ports 1 to 10 on the interface module.

To configure the untagged max-frame-size on a VLAN, enter a command such as the following at he Interface Configuration level.

```
BigIron RX(config-vlan-20)#
BigIron RX(config-vlan-20)#max-frame-size 5000
Please reload system!
BigIron RX(config-vlan-20)#
```

Syntax: max-frame-size <bytes>

The <frame-size> variable specifies the maximum frame size for each port that is connected the same PPCR as described in Table 47. Values can be from 64 to 9212 bytes. The default is 1518 bytes.

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 device 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 and 1492 for Ethernet SNAP packets. You can change the IP MTU globally or an individual ports. You can increase the IP MTU size to accommodate large packet sizes, such as jumbo packets, globally or on individual physical ports. However, IP MTU cannot be set higher than the maximum frame size, minus 18.

For jumbo packet, the device 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.

Configuration considerations for Increasing the IP MTU

Consider the following before configuring the maximum value to increase 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 on a port, minus 18 bytes. The 18 bytes is used for IP overhead, VLAN tagging, etc.
- When you increase the IP MTU size of a port, the increase uses system resources. Increase the IP MTU size only on the ports that need it. For example, if you have one port connected to a server that uses jumbo frames and two other ports connected to clients that can support the jumbo frames, increase the IP MTU only on those three ports. Leave the IP MTU size on the other ports at the default value (1500 bytes). Globally increase the IP MTU size only if needed.
- Use the same IP MTU size on all ports that will be supporting jumbo frames. If the device needs to fragment a jumbo frame (and the frame does not have the DF bit set), the device fragments the frame into 1500-byte fragments, even if the outbound port has a larger IP MTU. For example, if a port has an IP MTU setting of 8000 and receives an 8000-byte frame, then must forward the frame onto a port with an IP MTU of 4000, the device does not fragment the 8000-byte frame into two 4000-byte frames. Instead, the device fragments the 8000-byte frame into six fragments (five 1500-byte fragments and a final, smaller fragment.)

Globally changing the IP MTU

To globally enable jumbo support on all ports, enter commands such as the following.

```
BigIron RX(config)# ip mtu 5000
BigIron RX(config)# write memory
```

Syntax: [no] ip mtu <*bytes*>

The <*bytes*> parameter specifies the maximum number of bytes an Ethernet frame can have in order to be forwarded on a port. Enter 64 – 9212, but this value must be 18 bytes less than the value of the global maximum frame size.

NOTE
The BigIron RX will always use 22 Bytes less than the configured MTU in order to compensate for the 4Bytes required for VLAN tags. This is so if a packet is forwarded on both a tagged and untagged link within a VLAN, it will get through.

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
- **1492 bytes** – The maximum for SNAP encapsulation

NOTE
The IP MTU configured at the physical interface level takes precedence over the IP MTU configured at the global level for that physical interface.

To change the IP MTU for interface 1/5 to 1000, enter the following commands.

```
BigIron RX(config)# int e 1/5
BigIron RX(config-if-e10000-5)# ip mtu 1000
```

Syntax: [no] ip mtu <bytes>

The <bytes> parameter specifies the IP MTU. Ethernet II packets can hold IP packets from 572 – 1500 bytes long. Ethernet SNAP packets can hold IP packets from 572 – 1492 bytes long. However, the value of IP MTU on an interface cannot exceed the configured value of IP MTU for an interface, minus 18 bytes. The default IP MTU for Ethernet II packets is 1500. The default IP MTU for SNAP packets is 1492.

Changing the router ID

In most configurations, a device has multiple IP addresses, usually configured on different interfaces. As a result, a device'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 device by just one of the IP addresses configured on the device, regardless of the interfaces that connect the device 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 device 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 device. 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 cannot be in use on another device in the network.

NOTE
The device 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** CLI command at any CLI level.

To change the router ID, enter a command such as the following.

```
BigIron RX(config)# ip router-id 209.157.22.26
```

Syntax: 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, but do not specify an IP address in use by another device.

Specifying a single source interface for Telnet, TACACS/TACACS+, or RADIUS packets

When the device originates a Telnet, 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 device 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 device to use a single source interface for all Telnet, TACACS/TACACS+, or RADIUS packets, the device 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, TACACS/TACACS+, or RADIUS packets provides the following benefits:

- If your Telnet, 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 *Brocade* 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, 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, TACACS/TACACS+, or RADIUS packets. You can configure a source interface for one or more of these types of packets separately.

To specify an Ethernet or a loopback or virtual interface as the source for all TACACS/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/TACACS+ packets originated by the device.

The following sections show the syntax for specifying a single source IP address for Telnet, TACACS/TACACS+, and RADIUS packets.

Telnet packets

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

```
BigIron RX(config)# int loopback 2
BigIron RX(config-lbif-2)# ip address 10.0.0.2/24
BigIron RX(config-lbif-2)# exit
BigIron RX(config)# ip telnet source-interface loopback 2
```

The commands configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all Telnet packets from the device.

Syntax: ip telnet source-interface ethernet <slot/port> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number.

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 device.

```
BigIron RX(config)# interface ethernet 1/4
BigIron RX(config-if-e10000-1/4)# ip address 209.157.22.110/24
BigIron RX(config-if-e10000-1/4)# exit
BigIron RX(config)# ip telnet source-interface ethernet 1/4
```

TACACS/TACACS+ packets

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

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

The commands 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/TACACS+ packets from the device.

Syntax: ip tacacs source-interface ethernet <*slot/port*> | loopback <*num*> | ve <*num*>

The <*num*> parameter is a loopback interface or virtual interface number.

RADIUS packets

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.

```
BigIron RX(config)# int ve 1
BigIron RX(config-vif-1)# ip address 10.0.0.3/24
BigIron RX(config-vif-1)# exit
BigIron RX(config)# ip radius source-interface ve 1
```

The commands 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 device.

Syntax: ip radius source-interface ethernet <*slot/port*> | loopback <*num*> | ve <*num*>

The <*num*> parameter is a loopback interface or virtual interface number.

Configuring an interface as the source for Syslog packets

You can configure the device to use the lowest-numbered IPv4 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.

```
BigIron RX(config)# int ve 1
BigIron RX(config-vif-1)# ip address 10.0.0.4/24
BigIron RX(config-vif-1)# exit
BigIron RX(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.

NOTE
With this new command, the source ip of syslog is no longer controlled by the **snmp-server trap-source** command. In releases before 02.4.00, the **snmp-server trap-source** command controlled both SNMP and Syslog source IP.

IP fragmentation protection

Beginning with this release, IP packet filters on the device switches will drop undersized fragments and overlapping packet fragments to prevent tiny fragment attacks as explained in RFC 1858. When packets are fragmented on the network, the first fragment of a packet must be large enough to contain all the necessary header information. Fragments, once reassembled, must meet certain criteria before they are allowed to pass through the network. There are no CLI commands for this new security feature.

IP option attack protection

An attack on the network could be accomplished using the options field of an IP packet header. For example, the source routing option makes it possible for the sender to specify a route to follow.

To protect against attacks contained in the option field, device devices drop any IP packet that contains an option in its header, except for packets. IGMP packets are processes even if they contain IP options. If you want other packets that contain options in their headers to be processed, enter a command such as the following.

```
BigIron RX(config)#ip ip-option-process
```

Syntax: [no] ip ip-option-process

IP receive access list

The *IP receive access list* feature uses IPv4 ACLs to filter the packets intended for the management process to protect the management module from being overloaded with heavy traffic that was sent to one of the Layer 3 Switch IP interfaces. The feature applies to IPv4 unicast and multicast packets.

Configuring IP receive access list

IP receive access list is a global configuration command. Once it is applied, the command will be effective on all the management modules on the device. To configure the feature, do the following.

1. Create a numbered ACL that will be used as the IP receive ACL. This ACL can be a standard (1–99) or extended (100–199) ACL. Named ACLs are not supported.

   Example

   ```
   BigIron RX(config)# access-list 10 deny host 209.157.22.26 log
   BigIron RX(config)# access-list 10 deny 209.157.29.12 log
   BigIron RX(config)# access-list 10 deny host IPHost1 log
   BigIron RX(config)# access-list 10 permit any
   BigIron RX(config)# write memory
   ```

2. Configure ACL 10 as the IP receive access list by entering the following command.

   ```
   BigIron RX(config)# ip receive access-list 10
   ```

   Syntax: [no] ip receive access-list <*num*>

Specify an access list number for <*num*>.

The IP receive ACL is applied globally to all interfaces on the device.

Displaying IP receive access list

To determine if IP receive access list has been configured on the device, enter the following command.

```
BigIron RX# show access-list bindings
L4 configuration:

ip receive access-list 101
```

Configuring ARP parameters

Address Resolution Protocol (ARP) is a standard IP protocol that enables the device to obtain the MAC address of another device's interface when the device knows the IP address of the interface. ARP is enabled by default and cannot be disabled.

How ARP works

The device needs to know a destination's MAC address when forwarding traffic, because the device 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 device. The device can be the packet's final destination or the next-hop router toward the destination.

The device encapsulates IP packets in Layer 2 packets regardless of whether the ultimate destination is locally attached or is multiple router hops away. Since the device's IP route table and IP forwarding cache contain IP address information but not MAC address information, the device cannot forward IP packets based solely on the information in the route table or forwarding cache. The device 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 device 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 device 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 device does the following:

- First, the device 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 device 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 device 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 device 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 device, the device sends an ARP response containing its MAC address. The response is a unicast packet addressed directly to the device. The deviceplaces 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 device. A MAC broadcast is not routed to other networks. However, some routers, including the device, 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 182.

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 device knows of no route to the destination address), the router sends an ICMP Host Unreachable message to the source.

Rate limiting ARP packets

You can limit the number of ARP packets the device accepts during each second. By default, the software does not limit the number of ARP packets the device can receive. Since the device sends ARP packets to the CPU for processing, if a device in a busy network receives a high number of ARP packets in a short period of time, some CPU processing might be deferred while the CPU processes the ARP packets.

To prevent the CPU from becoming flooded by ARP packets in a busy network, you can restrict the number of ARP packets the device will accept each second. When you configure an ARP rate limit, the device accepts up to the maximum number of packets you specify, but drops additional ARP packets received during the one-second interval. When a new one-second interval starts, the counter restarts at zero, so the device again accepts up to the maximum number of ARP packets you specified, but drops additional packets received within the interval.

To limit the number of ARP packets the device will accept each second, enter a command such as the following at the global CONFIG level of the CLI.

```
BigIron RX(config)# arp-port-rate-limit 100
```

This command configures the device to accept up to 100 ARP packets each second. If the device receives more than 100 ARP packets during a one-second interval, the device drops the additional ARP packets during the remainder of that one-second interval.

Syntax: [no] arp-port- rate-limit <*num*>

The <*num*> parameter specifies the number of ARP packets and can be from 0 – 30,000. If you specify 0, the device will not accept any ARP packets.

Applying a rate limit to ARP packets on an interface

To prevent the CPU from becoming flooded by ARP packets in a busy network, you can restrict the number of ARP packets an interface will accept each second. When ARP rate limit is configured on an interface, the interface will accept up to the maximum number of packets you specify, but drops additional ARP packets received during the one-second interval. When a new one-second interval starts, the counter restarts at zero, so the interface again accepts up to the maximum number of ARP packets you specified, but drops additional packets received within the interval. This feature is disabled by default.

Configuration notes

- When configuring ARP rate limiting globally, interfcae level ARP rate-limiting gets removed.
- The interface level configuration overrides the global configuration for a specific port.
- The command is supported on Layer 3 Switches only.
- There is no default value for <rate>. Enter 0–30,000.
- If the value of <rate> is entered as 0, the interface will stop processing ARP packets immediately.
- You can go to interface trunk mode to configure the ARP port rate limit. When configured over trunk interface (i.e. on the lead port) the same limit will be configured on each and every port in the trunk.
- ARP rate limiting is only supported on physical interfaces (virtual interfaces (ve) are not supported).

Setting the rate limit to ARP packets on an interface

You can limit the number of ARP packets the device will accept each second by entering the **arp-port-rate-limit** command. However, if you want to apply a limit on the rate that ARP packets flow on an interface of a Layer 3 Switch, enter a command such as the following.

```
BigIron RX(config)#interface ethernet 1/4
BigIron RX(config-vif-10)#arp-port-rate-limit 2000
```

Syntax: [no] arp-port-rate-limit <*rate*>

There is no default value for <*rate*>. Enter 0–30,000.

Displaying the rate limit for ARP packets

To determine how many ARP packets were dropped by an interface due to the configured rate limit for ARP packets, enter a command such as the following.

```
LP-1#show ip traffic arp
ARP Statistics
  1400 total recv, 1400 req recv, 0 req sent
  0 pending drop, 0 invalid source, 0 invalid dest

ARP Rate Limiting Statistics
Interface        Received        Processed        Dropped(Rate-limted)
ethernet1/1          184200           700              183500
ethernet1/2          0                0                0
ethernet1/3          0                0                0
ethernet1/4          184200           700              183500
```

The example above displays the LP processed 50 packets every second and dropped any addtional packets.

Syntax: show ip traffic arp

This column...	Displays...
Interface	The interface on the device.
Received	Number of ARP packets received by the interface.
Processed	Number of ARP packets processed by the interface.
Dropped (Rate-limited)	Number of ARP packets dropped by the interface.

Clearing the rate limit for ARP packets

To clear the ARP port rate limit data on every port of the LP, enter a command such as the following.

```
LP-1# clear ip traffic arp
```

Changing the ARP aging period

When the device places an entry in the ARP cache, the device 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 device, 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.

```
BigIron RX(config)# ip arp-age 20
```

Syntax: 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.

```
BigIron RX(config-if-e1000-1/1)# ip arp-age 30
```

Enabling proxy ARP

Proxy ARP allows the device 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 device connected to two subnets, 10.10.10.0/24 and 20.20.20.0/24, the device 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.

```
BigIron RX(config)# ip proxy-arp
```

To again disable IP proxy ARP, enter the following command.

```
BigIron RX(config)# no ip proxy-arp
```

Syntax: [no] ip proxy-arp

Creating static ARP entries

The device 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 device, 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 *Brocade* 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 maximum number of entries the static ARP table can hold" on page 184.

To display the ARP cache and static ARP table, see the following:

- To display the ARP table, refer to "Displaying the ARP cache" on page 217.
- To display the static ARP table, refer to "Displaying the static ARP table" on page 218.

To create a static ARP entry for a static MAC entry, enter a command such as the following.

```
BigIron RX(config)# arp 1 192.53.4.2 1245.7654.2348 e 1/2
```

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 port 1/2 of the device.

Syntax: arp <*ip-addr*> <*mac-addr*> ethernet <*slot/port*>

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** <*slot/port*> command specifies the port number attached to the device that has the MAC address of the entry.

The **arp** command allows you to specify only one port number. To create a static ARP entry for a static MAC entry that is associated with multiple ports, specify the first (lowest-numbered) port associated with the static MAC entry.

Changing the maximum number of entries the static ARP table can hold

The default number of entries in the static ARP table on the device are as follows:

- Default maximum: 8192
- Configurable maximum: 65536

NOTE
You must save the configuration to the startup configuration file and reload the software after changing the static ARP table size to place the change into effect.

NOTE
The basic procedure for changing the static ARP table size is the same as the procedure for changing other configurable cache or table sizes. Refer to "Displaying and modifying system parameter default settings" on page 127.

To increase the maximum number of entries in the static ARP table you can configure, enter commands such as the following at the global CONFIG level of the CLI.

```
BigIron RX(config)# system-max ip-static-arp 4000
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# reload
```

Syntax: system-max ip-static-arp <*num*>

The <*num*> parameter indicates the maximum number of static ARP entries: 2048 - 4096 (default: 2048).

As of IronWare release 02.4.00 the maximum number of static ARP entries is 16384 (default: 2048).

NOTE
As of release 2.4.00, the **system-max static-arp** command no longer affects memory allocation for static ARPs. Instead, the BigIron RX dynamically allocates memory for static-arp entries as required and this is only limited by the memory allocation for all ARP entries, specified by the system-max ip-arp command.

Creating a floating static ARP entry

Beginning with release 02.5.00, you can create a static ARP entry without port assignments.

When a floating static ARP entry (Static ARP entry without the outgoing interface defined) is added to the ARP Inspection table, the mapping is checked against the current static ARP table. If an 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.

```
BigIron RX(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: arp <*ip-add*> <*mac-addr*>

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.

Static route ARP validation check

Beginning with release 02.5.00, you can configure the device to perform validation checks on the destination MAC address, the sender and target IP addresses, and the source MAC address.

You can enable ARP validation check on the global basis. When feature is enabled, the static route will only be installed when the next hop ARP has been resolved.

Configuring an ARP validation check

To enable the ARP validation check globally, enter a command such as the following.

```
BigIron RX(config)#ip route validate-nexthop-arp
```

Syntax: [no] ip route validate-nexthop-arp

Use the **no** form of the command to disable the ARP validation feature. When ARP validation is disabled, the static route will be installed without checking the validity of the next hop.

Enabling the next hop validate ARP timer

The next hop validate ARP timer works only on the ARP entries created when the ARP validation check feature has been enabled. The timer is used to age out the ARP entries when the next hop goes down. All other ARP entries in the system, which are NOT created due to static routes, follow the normal ARP age timer with default value of 3 minutes.

Use the ARP validation timer to reduce the response time where the static route with the next hop down can be replaced quickly with a route with active next hop.

To set the ARP validation timer to 30 seconds, enter commnads such as the following.

```
BigIron RX(config)#ip route validate-nexthop-arp
BigIron RX(config)#ip route validate-nexthop-arp timer 30
```

Syntax: [no] ip route validate-nexthop-arp timer <*value*>

The default is 200 seconds.

The value parameter speocifies the amount of time before a nexthop down is replaced by an active nexthop. Possible values are10-200 seconds.

Use the **no** form of the command to disable the validation timer.

Displaying the routes waiting for the next hop ARP to resolve

Use the following command to display which routes are waiting for the nexthop ARP to be resolved.

```
BigIron RX# show ip static route
IP Static Routing Table - 2 entries:
Type Codes: '*' - Installed, '+' - Waiting for ARP resolution
IP Prefix          Next Hop        Interface  Dis/Metric/Tag
*10.0.0.0/8        10.43.14.1                 1/1/0
+20.1.1.0/24       12.1.1.2                   1/1/0
*20.1.1.0/24       12.1.1.6                   1/1/0
+20.1.1.0/24       12.1.1.7                   5/1/0
20.1.1.0/24         10.43.14.1                   10/1/0
```

Displaying ARP

When the next hop entry is a staic route, enter the following command to display the route and the timer value.

```
BigIron RX# show arp 10.43.14.1
Total number of ARP entries: 1
      IP Address          MAC Address      Type     Age      Port       Status
1     10.43.14.1           00ab.cdef.0100   Dynamic  5         mgmt1    Valid
     ARP Debug Info
      ArpIndex 0 InstId 16840 OutInt 2048 Vlan:0
      HwMacIndex 0x0000ffff Router 0 PktCount 0
       NumReq 0 ReplyTimeout 100
```

For additional information on the command syntax, refer to the syntax of the show arp command under “Displaying the ARP cache” on page 217.

Configuring forwarding parameters

The following configurable parameters control the forwarding behavior of the device:

- 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 device.

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 device 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.

```
BigIron RX(config)# ip ttl 25
```

Syntax: 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.

```
BigIron RX(config)# ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

Brocade 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.

```
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(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 device supports both types of IP source routing:

- **Strict source routing** – requires the packet to pass through only the listed routers. If the device receives a strict source-routed packet but cannot reach the next hop interface specified by the packet, the device discards the packet and sends an ICMP Source-Route-Failure message to the sender.

  NOTE
  The device allows you to disable sending of the Source-Route-Failure messages. Refer to "Disabling ICMP messages" on page 188.

- **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 device 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.

```
BigIron RX(config)# no ip source-route
```

Syntax: [no] ip source-route

To re-enable forwarding of source-routed packets, enter the following command.

```
BigIron RX(config)# ip source-route
```

Enabling support for zero-based IP subnet broadcasts

By default, the device treats IP packets with all ones in the host portion of the address as IP broadcast packets. For example, the device 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 device).

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 device to treat IP packets with all zeros in the host portion of the destination IP address as broadcast packets.

NOTE
When you enable the device for zero-based subnet broadcasts, the device still treats IP packets with all ones the host portion as IP subnet broadcasts too. Thus, the device 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 device for zero-based IP subnet broadcasts in addition to ones-based IP subnet broadcasts, enter the following command.

```
BigIron RX(config)# ip broadcast-zero
```

Syntax: [no] ip broadcast-zero

Disabling ICMP messages

The device 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 device replies to IP pings from other IP devices.

- **Destination Unreachable messages** – If the device receives an IP packet that it cannot deliver to its destination, the device discards the packet and sends a message back to the device that sent the packet. The message informs the device that the destination cannot be reached by the device.

Disabling replies to broadcast ping requests

By default, the device 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.

```
BigIron RX(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.

```
BigIron RX(config)# ip icmp echo broadcast-request
```

Disabling ICMP destination unreachable messages

By default, when the device 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 device's response to the following types of ICMP Unreachable messages:

- **Administration** – The packet was dropped by the *Brocade* device due to a filter or ACL configured on the device.
- **Fragmentation-needed** – The packet has the Don't Fragment bit set in the IP Flag field, but the device cannot forward the packet without fragmenting it.
- **Host** – The destination network or subnet of the packet is directly connected to the device, but the host specified in the destination IP address of the packet is not on the network.
- **Network** – The device 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 device, 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 device from sending these types of ICMP messages on an individual basis.

NOTE
Disabling an ICMP unreachable message type does not change the device'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.

```
BigIron RX(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 Don't-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.

```
BigIron RX(config)# no ip icmp unreachable host
BigIron RX(config)# no ip icmp unreachable network
```

If you have disabled all ICMP Unreachable message types but you want to re-enable certain types, you can do so entering commands such as the following.

```
BigIron RX(config)# ip icmp unreachable host
BigIron RX(config)# ip icmp unreachable network
```

The commands shown above 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 device 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
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.

```
BigIron RX(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.

```
BigIron RX(config)# int e 3/11
BigIron RX(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 device automatically creates a route for the network the interface is in.
- **RIP** – If RIP is enabled, the device can learn about routes from the advertisements other RIP routers send to the device. If the route has a lower administrative distance than any other routes from different sources to the same destination, the device places the route in the IP route table.
- **OSPF** – See RIP, but substitute “OSPF” for “RIP”.
- **BGP4** – See RIP, but substitute “BGP4” for “RIP”.
- **Default network route** – A statically configured default route that the device uses if other default routes to the destination are not available. Refer to “Configuring a default network route” on page 200.
- **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 device interface through which you want the device 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 device drops traffic forwarded to the null interface.

The following parameters are optional:

- **The route's metric** – The value the device 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 device 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 device 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 device 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 device can load balance traffic to the routes' destination. For information about IP load balancing, refer to "Configuring IP load sharing" on page 201.
- **Path redundancy** – When you add multiple static IP routes for the same destination, but give the routes different metrics or administrative distances, the device uses the route with the lowest administrative distance by default, but uses another route to the same destination of the first route becomes unavailable.

See the following sections for examples and configuration information:

- "Configuring load balancing and redundancy using multiple static routes to the same destination" on page 196
- "Configuring standard static IP routes and interface or null static routes to the same destination" on page 197

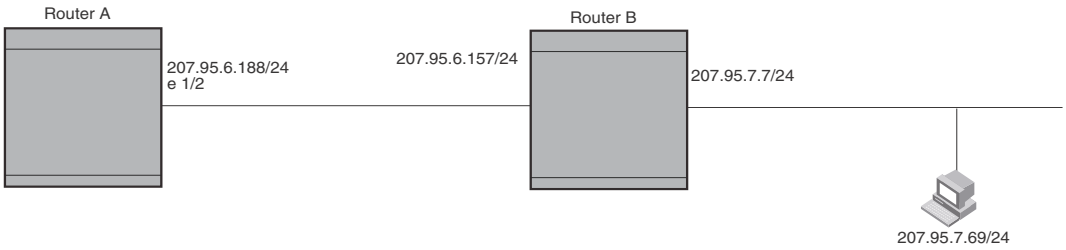
Static route states follow port states

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 device to adjust to changes in network topology. The device does not continue trying to use routes on unavailable paths but instead uses routes only when their paths are available.

Figure 10 shows a network containing a static route. The static route is configured on Router A, as shown in the CLI following the figure.

FIGURE 10 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.

```
BigIron RX(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 device interface through which the device can reach the route. The device 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.

```
BigIron RX(config)# ip route 192.0.0.0 255.0.0.0 195.1.1.1
```

To configure a default route, enter the following.

BigIron RX(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.

```
BigIron RX(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 device 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.

```
BigIron RX(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> | ve <num> [<metric>] [tag <num>] [distance <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 / 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 device. The <*num*> parameter is a virtual interface number. The <slot/port> is the port's number of the device. If you specify an Ethernet port, the device 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 device 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 device 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 device will replace the static route if it receives a route with a lower administrative distance. Refer to "Changing administrative distances" on page 765 for a list of the default administrative distances for all types of routes.

Configuring a "null" route

You can configure the device 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 device receives a packet destined for the address, the device 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.

```
BigIron RX(config)# ip route 209.157.22.0 255.255.255.0 null0
BigIron RX(config)# write memory
```

Syntax: 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 device 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 device's CPU.

Hardware dropping for IP traffic sent to the null0 interface is supported.

You can optionally configure the device to drop traffic sent to the default IP route address in hardware. To do this, enter the following commands.

```
BigIron RX(config)# ip route 0.0.0.0 0.0.0.0 null0
BigIron RX(config)# ip hw-drop-on-def-route
```

Syntax: [no] ip hw-drop-on-def-route

Configuring the device 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:

```
BigIron RX(config)# ip dr-aggregate
```

Syntax: ip dr-aggregate

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:

```
BigIron RX(config)#ip route 192.122.12.1 255.255.255.0 192.122.1.1 tag 20
```

Syntax: 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.

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 device 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 device alternates between the two routes. For information about IP load balancing, refer to "Configuring IP load sharing" on page 201.
- **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 device will always use the route with the lowest metric. If this route becomes unavailable, the device will fail over to the static route with the next-lowest metric, and so on.

NOTE
You also can bias the device 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. For a list of the default administrative distances, refer to "Changing administrative distances" on page 765.

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.

```
BigIron RX(config)# ip route 192.128.2.69 255.255.255.0 209.157.22.1
BigIron RX(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 device uses the route with the lowest metric if the route is available.

```
BigIron RX(config)# ip route 192.128.2.69 255.255.255.0 209.157.22.1
BigIron RX(config)# ip route 192.128.2.69 255.255.255.0 192.111.10.1 2
BigIron RX(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 193.

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 device has multiple routes to the same destination, the device 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 device 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 device 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 device 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 11 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 device always prefers the static route with the lower metric. In this example, the device 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 device sends traffic to the null route instead.

FIGURE 11 Standard and null static routes to the same destination network

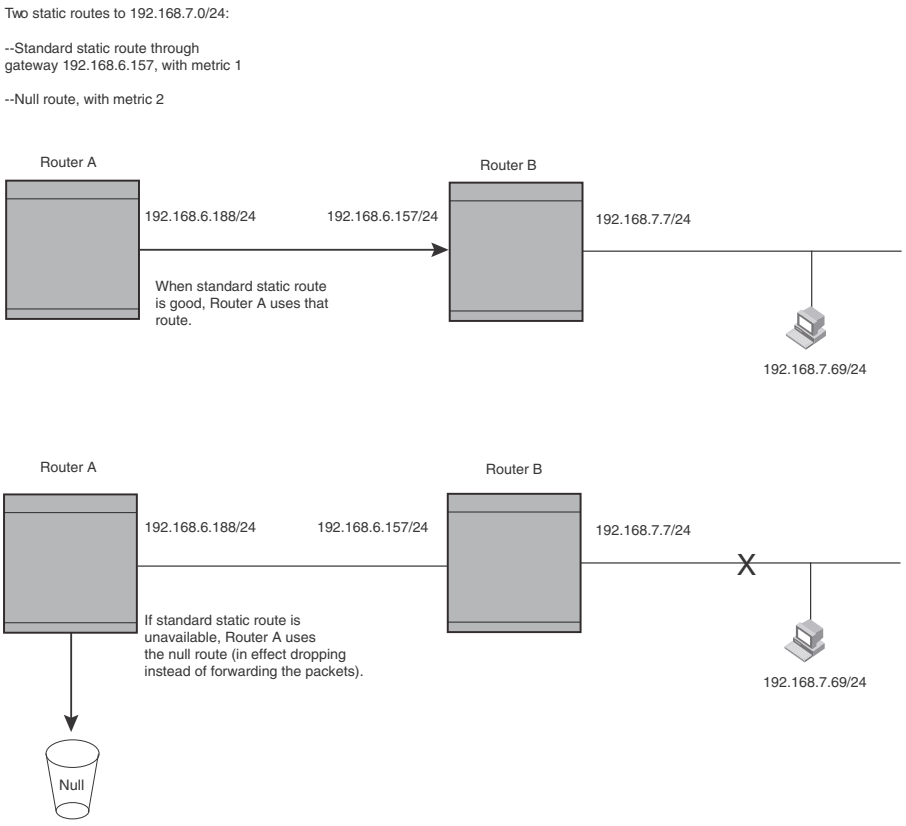
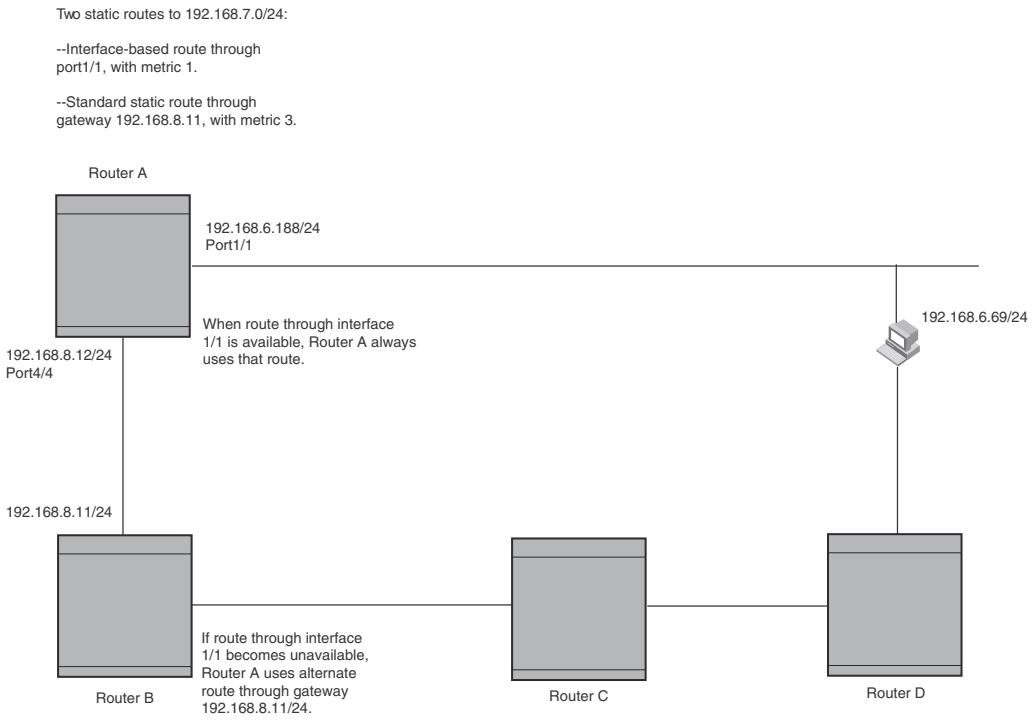


Figure 12 shows another example of two static routes. A standard static route and an interface-based static route are configured for destination network 192.168.6.0/24. The interface-based static route has a lower metric than the standard static route. As a result, the device always prefers the interface-based route when the route is available. However, if the interface-based route becomes unavailable, the device still forwards the traffic toward the destination using an alternate route through gateway 192.168.8.11/24.

FIGURE 12 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 11 on page 198, enter commands such as the following.

```
BigIron RX(config)# ip route 192.168.7.0/24 192.168.6.157/24 1
BigIron RX(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 device 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 to "Configuring a static IP route" on page 193.

To configure a standard static route and an interface-based route to the same destination, enter commands such as the following.

```
BigIron RX(config)# ip route 192.168.6.0/24 ethernet 1/1 1
BigIron RX(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 device to always prefer this route when it is available. If the route becomes unavailable, the device uses an alternate route through the next-hop gateway 192.168.8.11/24.

Configuring a default network route

The device 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 device 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 device 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.

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.

```
BigIron RX(config)# ip default-network 209.157.22.0
BigIron RX(config)# write memory
```

Syntax: 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.

```
BigIron RX(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 device 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 device uses ***IP load sharing*** to select a path to the destination.[1]

IP load sharing is based on the destination address of the traffic. device supports load sharing based on individual host addresses or on network addresses.

You can enable a device 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.

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 device 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 device 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 device:

- **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 device 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 device 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 196.
- **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 48 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 device, load sharing for static IP routes, RIP routes, OSPF routes, and BGP4 routes also is enabled by default.

TABLE 48 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 204
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 204

TABLE 48 Default load sharing parameters for route sources (Continued)

Route source	Default maximum number of paths	Maximum number of paths	See...
OSPF	4	8	page 204
BGP4	1	4	page 789

How IP load sharing works

On the device, IP load sharing (also known as ECMP 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, and IP protocol. 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 device 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 device 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.

BigIron RX(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.

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 204.)

Default route ECMP

On the BigIron RX, IP load sharing (also known as ECMP 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, and IP protocol. This hash is used to select one of the paths.

If there are multiple next-hop routers for the default route in the IPv4 routing table, routed packets on the default route would be automatically load-balanced among these next-hops through a hashing formula, calculated based on (IPv4 Destination Address, IPv4 Source Address, IPv4 Source Port, IPv4 Destination Port, DA-MAC, and SA-MAC) of the packets received. This feature allows for load distribution of traffic among the available default route next-hops.

NOTE
This feature is currently not applicable to IPv6 traffic.

To specify the ECMP default route, enter a command such as the following.

```
BigIron RX(config)# ip load-sharing default-route
```

Syntax: [no] ip load-sharing [<*num*> I <*default-route*>]

The <*num*> parameter specifies the number of paths and can be from 2 – 8.

The <*default-router*> parameter specifies the ECMP load sharing.

DIsplaying the ECMP load sharing

Use the **show run** command to display the ECMP load sharing.

```
BigIron RX(config)#show run
========show run ====================
!
logging console
hostname RW
ip route 0.0.0.0/0 100.1.1.2
ip route 0.0.0.0/0 100.1.2.2
ip route 0.0.0.0/0 100.1.3.2
ip route 0.0.0.0/0 100.1.4.2
ip route 10.0.0.0/8 10.43.2.1
ip route 40.0.0.0/24 100.1.1.2
ip load-sharing default-route
```

Use the **show ip route** command to display the traffic that will now be sent over all 4 links load balanced instead of being on only 1 link.

```
BigIron RX#show ip route
Total number of IP routes: 9
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
Destination Gateway Port Cost Type
1 0.0.0.0/0 100.1.1.2 eth 7/1 1/1 S
0.0.0.0/0 100.1.2.2 eth 7/2 1/1 S
0.0.0.0/0 100.1.3.2 eth 7/3 1/1 S
0.0.0.0/0 100.1.4.2 eth 7/4 1/1 S
2 10.0.0.0/8 10.43.2.1 mgmt 1 1/1 S
3 10.43.2.0/24 DIRECT mgmt 1 0/0 D
4 40.0.0.0/24 100.1.1.2 eth 7/1 1/1 S
5 70.1.1.0/24 DIRECT eth 7/9 0/0 D
6 100.1.1.0/24 DIRECT eth 7/1 0/0 D
7 100.1.2.0/24 DIRECT eth 7/2 0/0 D
8 100.1.3.0/24 DIRECT eth 7/3 0/0 D
9 100.1.4.0/24 DIRECT eth 7/4 0/0 D
```

IP receive access list

The *IP receive access list* feature uses IPv4 ACLs to filter the packets intended for the management process to protect the management module from being overloaded with heavy traffic that was sent to one of the Layer 3 Switch IP interfaces. The feature applies to IPv4 unicast and multicast packets.

Configuring IP receive access list

IP receive access list is a global configuration command. Once it is applied, the command will be effective on all the management modules on the device. To configure the feature, do the following.

1. Create a numbered ACL that will be used as the IP receive ACL. This ACL can be a standard (1–99) or extended (100–199) ACL. Named ACLs are not supported.

   Example

   ```
   BigIron RX(config)# access-list 10 deny host 209.157.22.26 log
   BigIron RX(config)# access-list 10 deny 209.157.29.12 log
   BigIron RX(config)# access-list 10 deny host IPHost1 log
   BigIron RX(config)# access-list 10 permit any
   BigIron RX(config)# write memory
   ```

2. Configure ACL 10 as the IP receive access list by entering the following command.

   ```
   BigIron RX(config)# ip receive access-list 10
   ```

   Syntax: [no] ip receive access-list <*num*>

Specify an access list number for <*num*>.

The IP receive ACL is applied globally to all interfaces on the device.

Displaying IP receive access list

To determine if IP receive access list has been configured on the device, enter the following command.

```
BigIron RX# show access-list bindings
L4 configuration:

ip receive access-list 101
```

Configuring IRDP

The device 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 port:

- 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 device periodically sends Router Advertisement messages out the IP interfaces on which the feature is enabled. The messages advertise the device's IP addresses to directly attached hosts who listen for the messages. In addition, hosts can be configured to query the device 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 device responds to the Router Solicitation messages. Some clients interpret this response to mean that the device is the default gateway. If another router is actually the default gateway for these clients, leave IRDP disabled on the device.

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 using:

- **Packet type** – The device 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 device sends the Router Advertisement messages every 450 – 600 seconds by default. The time within this interval that the device 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 device 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.

```
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 1/3
BigIron RX(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 device uses to send Router Advertisement.

- **broadcast** – The device sends Router Advertisement as IP broadcasts. This is the default.
- **multicast** – The device 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 device should consider the advertisement to be valid. When a host receives a new Router Advertisement message from the device, the host resets the hold time for the device 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 device 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 device 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 device. 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 device 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 device forwards client requests for any of the application ports the device 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 device 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/DHCP is enabled by default. If you are configuring the device to forward BootP/DHCP requests, refer to "Configuring BootP/DHCP forwarding parameters" on page 211.

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 device is not also disabled.

Enabling forwarding for a UDP application

If you want the device to forward client requests for UDP applications that the device 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 device cannot forward the requests unless you configure the helper address. Refer to "Configuring an IP helper address" on page 212.

To enable the forwarding of SNMP trap broadcasts, enter the following command.

```
BigIron RX(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.

```
BigIron RX(config)# no ip forward-protocol udp snmp
```

Syntax: [no] ip forward-protocol udp snmp

This command disables forwarding of SNMP requests to the helper addresses configured on device 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.

```
BigIron RX(config)# interface e 1/2
BigIron RX(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 device is enabled to forward, the device forwards the client's request to the server.

Syntax: 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/DHCP forwarding parameters

Beginning with release 02.7.00, the DHCP relay will allow for IP address grants that do not match the subnets configured on the interface that the DHCP request was received. A host on an IP network can use BootP/DHCP to obtain its IP address from a BootP/DHCP server. To obtain the address, the client sends a BootP/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 device or other IP routers.

When the BootP/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 device does not forward the request.

You can configure the device to forward BootP/DHCP requests. To do so, configure a helper address on the interface that receives the client requests, and specify the BootP/DHCP server's IP address as the address you are helping the BootP/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/DHCP request, does not have to match the subnet of the IP address given by the DHCP server.

BootP/DHCP forwarding parameters

The following parameters control the device's forwarding of BootP/DHCP requests:

- **Helper address** – The BootP/DHCP server's IP address. You must configure the helper address on the interface that receives the BootP/DHCP requests from the client. The device cannot forward a request to the server unless you configure a helper address for the server.
- **Gateway address** – The device places the IP address of the interface that received the BootP/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 device 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 device to use.
- **Hop Count** – Each router that forwards a BootP/DHCP packet increments the hop count by 1. Routers also discard a forwarded BootP/DHCP request instead of forwarding the request if the hop count is greater than the maximum number of BootP/DHCP hops allows by the router. By default, the device forwards a BootP/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 device will allow to a value from 1 – 15.

  NOTE
  The BootP/DHCP hop count is not the TTL parameter.

Configuring an IP helper address

The procedure for configuring a helper address for BootP/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 210 .

Changing the IP address used for stamping BootP/DHCP requests

When the device forwards a BootP/DHCP request, the device "stamps" the Gateway Address field. The default value the device uses to stamp the packet is the lowest-numbered IP address configured on the interface that received the request.

The BootP/DHCP stamp address is an interface parameter. Change the parameter on the interface that is connected to the BootP/DHCP client.

To change the IP address used for stamping BootP/DHCP requests received on interface 1/1, enter commands such as the following.

```
BigIron RX(config)# int e 1/1
BigIron RX(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/DHCP stamp address for requests received on port 1/1 to 192.157.22.26. The device will place this IP address in the Gateway Address field of BootP/DHCP requests that the device receives on port 1/1 and forwards to the BootP/DHCP server.

Syntax: ip bootp-gateway <*ip-addr*>

Changing the maximum number of hops to a BootP relay server

Each BootP/DHCP request includes a field Hop Count field. The Hop Count field indicates how many routers the request has passed through. When the device receives a BootP/DHCP request, the device 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 device allows, the device increments the hop count by one and forwards the request.
- If the hop count is greater than the maximum hop count the device allows, the device discards the request.

NOTE
The BootP/DHCP hop count is not the TTL parameter.

To modify the maximum number of BootP/DHCP hops, enter the following command.

```
BigIron RX(config)# bootp-relay-max-hops 10
```

This command allows the device to forward BootP/DHCP requests that have passed through up to ten previous hops before reaching the device.

Syntax: bootp-relay-max-hops <*1-15*>

Default: 4

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 213.
- **IP interfaces** – refer to "Displaying IP interface information" on page 215.
- **ARP entries** – refer to "Displaying ARP entries" on page 217.
- **Static ARP entries** – refer to "Displaying ARP entries" on page 217.
- **IP forwarding cache** – refer to "Displaying the forwarding cache" on page 219.
- **IP route table** – refer to "Displaying the IP route table" on page 220.
- **IP traffic statistics** – refer to "Displaying IP traffic statistics" on page 223.

The sections below describe how to display this information.

In addition to the information described below, you can display the following IP information. This information is described in other parts of this guide:

- **RIP information** – refer to "Displaying RIP filters" on page 672.
- **OSPF information** – refer to "Displaying OSPF information" on page 717.
- **BGP4 information** – refer to "Displaying BGP4 information" on page 822.
- **DVMRP information** – refer to "Displaying information about an upstream neighbor device" on page 651
- **PIM information** – refer to "Displaying PIM Sparse configuration information and statistics" on page 610.
- **VRRP or VRRPE information** – refer to "Displaying VRRP and VRRPE information" on page 456.

Displaying global IP configuration information

To display IP configuration information, enter the following command at any CLI level.

```
BigIron RX> show ip

Global Settings
  ttl: 64, arp-age: 10, bootp-relay-max-hops: 4
  router-id : 207.95.11.128
  enabled : UDP-Broadcast-Forwarding  IRDP  Proxy-ARP  OSPF
  disabled: BGP4 Load-Sharing  RIP  DVMRP  FSRP  VRRP
Static Routes
  Index   IP Address        Subnet Mask       Next Hop Router   Metric Distance
  1       0.0.0.0           0.0.0.0           209.157.23.2      1      1
Policies
  Index   Action   Source           Destination      Protocol   Port  Operator
  1       deny     209.157.22.34    209.157.22.26    tcp        http  =
  64      permit   any              any
```

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 49 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 device. If the packet's TTL value is higher than the value specified in this field, the *Brocade* router drops the packet. To change the maximum TTL, refer to "Changing the TTL threshold" on page 186.
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 182.
bootp-relay-max-hops	The maximum number of hops away a BootP server can be located from the *Brocade* 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 212.
router-id	The 32-bit number that uniquely identifies the *Brocade* 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 174.
enabled	The IP-related protocols that are enabled on the router.
disabled	The IP-related protocols that are disabled on the router.
Static routes	
Index	The row number of this entry in the IP route table.
IP Address	The IP address of the route's destination.
Subnet Mask	The network mask for the IP address.
Next Hop Router	The IP address of the router interface to which the *Brocade* router sends packets for the route.
Metric	The cost of the route. Usually, the metric represents the number of hops to the destination.
Distance	The administrative distance of the route. The default administrative distance for static IP routes in *Brocade* routers is 1. To list the default administrative distances for all types of routes or to change the administrative distance of a static route, refer to "Changing administrative distances" on page 765.
Policies	
Index	The policy number. This is the number you assigned the policy when you configured it.
Action	The action the router takes if a packet matches the comparison values in the policy. The action can be one of the following: • **deny** – The router drops packets that match this policy. • **permit** – The router forwards packets that match this policy.
Source	The source IP address the policy matches.

TABLE 49 CLI display of global IP configuration information (Continued)

This field...	Displays...
Destination	The destination IP address the policy matches.
Protocol	The IP protocol the policy matches. The protocol can be one of the following: • ICMP • IGMP • IGRP • OSPF • TCP • UDP
Port	The Layer 4 TCP or UDP port the policy checks for in packets. The port can be displayed by its number or, for port types the router recognizes, by the well-known name. For example, TCP port 80 can be displayed as HTTP. **NOTE:** This field applies only if the IP protocol is TCP or UDP.
Operator	The comparison operator for TCP or UDP port names or numbers. **NOTE:** This field applies only if the IP protocol is TCP or UDP.

Displaying IP interface information

To display IP interface information, enter the following command at any CLI level.

```
BigIron RX(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 50 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”.

TABLE 50 CLI display of interface IP configuration information (Continued)

This field...	Displays...
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".

To display detailed IP information for a specific interface, enter a command such as the following.

```
BigIron RX# show ip interface ethernet 1/1
Interface Ethernet 1/1
  port state: UP
  ip address: 192.168.9.51      subnet mask: 255.255.255.0
  encapsulation: ETHERNET, mtu: 1500, metric: 1
  directed-broadcast-forwarding: disabled
  proxy-arp: disabled
  ip arp-age:  10 minutes
  Ip Flow switching is disabled
  No Helper Addresses are configured.
  No inbound ip access-list is set
  No outgoing ip access-list is set
```

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.

```
BigIron RX(config)# ip show-portname
```

This command is applied globally to all interfaces on the device.

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.

```
BigIron RX># 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 device. 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.

```
BigIron RX# 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
```

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 "f"s and "0"s, where "f"s 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 51 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 device learned the entry from an incoming packet. • Static – The device loaded the entry from the static ARP table when the device for the entry was connected to the device.
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 213. To change the ARP aging interval, refer to "Changing the ARP aging period" on page 182. **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.

```
BigIron RX# show ip static-arp

Static ARP table size: 512, configurable from 512 to 1024
  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
```

This example shows two static entries. Note that since you specify an entry's index number when you create the entry, it is possible for the range of index numbers to have gaps, as shown in this example. The entry number you assign to a static ARP entry is not related to the entry numbers in the ARP cache.

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, refer to the syntax of the **show arp** command under "Displaying the ARP cache" on page 217.

TABLE 52 CLI display of static ARP table

This field...	Displays...
Static ARP table size	The maximum number of static entries that can be configured on the device using the current memory allocation. The range of valid memory allocations for static ARP entries is listed after the current allocation. To change the memory allocation for static ARP entries, refer to "Changing the maximum number of entries the static ARP table can hold" on page 184.
Index	The number of this entry in the table. You specify the entry number when you create the entry.

TABLE 52 CLI display of static ARP table (Continued)

This field...	Displays...
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Port	The port attached to the device the entry is for.

Displaying the forwarding cache

To display the IP Forwarding Cache for directly connected hosts, enter the following command.

```
BigIron RX> 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**.

```
BigIron RX>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.

```
BigIron RX>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 53 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 *Brocade* 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 *Brocade* 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.

Displaying the IP route table

To display the IP route table, enter the following command at any CLI level.

```
BigIron RX> show ip route
Total number of IP routes: 514
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
Destination       Gateway           Port   Cost   Type
1.1.0.0           99.1.1.2          1/1    2      R
1.2.0.0           99.1.1.2          1/1    2      R
1.3.0.0           99.1.1.2          1/1    2      R
1.4.0.0           99.1.1.2          1/1    2      R
1.5.0.0           99.1.1.2          1/1    2      R
1.6.0.0           99.1.1.2          1/1    2      R
1.7.0.0           99.1.1.2          1/1    2      R
1.8.0.0           99.1.1.2          1/1    2      R
1.9.0.0           99.1.1.2          1/1    2      R
1.10.0.0          99.1.1.2          1/1    2      S
```

Beginning with release 02.4.00, the **show ip route** command has been enhanced to include the elapse time since an IP route was installed.

```
BigIron RX(config)#show ip route
Total number of IP routes: 2
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
Uptime - Days:Hours:Minutes:Seconds
        Destination        Gateway         Port        Cost     Type Uptime
1       10.0.0.0/8         10.43.1.1       mgmt 1      1/1      S    2:23:0:16
2       10.43.1.0/24       DIRECT          mgmt 1      0/0      D    2:23:0:18
```

Syntax: show ip route <*num*> | [<*ip-addr*> [<*ip-mask*>] [debug | detail | longer]] | connected | bgp | isis | ospf | rip | static | summary] [| 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 device.

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 **summary** option displays a summary of the information in the IP route table.

The default routes are displayed first.

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 device.

```
BigIron RX(config)# show ip route connected
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
      Destination       Gateway           Port   Cost   Type
      209.157.22.0      0.0.0.0           4/11   1      D
```

Notice that the route displayed in this example has "D" in the Type field, indicating the route is to a directly connected device.

Here is an example of how to use the **static** option. To display only the static IP routes.

```
BigIron RX(config)# show ip route static
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
      Destination       Gateway           Port   Cost   Type
      192.144.33.11     209.157.22.12     1/1    2      S
```

Notice that the route displayed in this example has "S" in the Type field, indicating the route is static.

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.

```
BigIron RX(config)# show ip route 209.159.0.0/16 longer
Starting index: 1 B:BGP D:Directly-Connected R:RIP S:Static O:OSPF
Destination NetMask Gateway Port Cost Type

52 209.159.38.0 255.255.255.0 207.95.6.101 1/1 1 S
53 209.159.39.0 255.255.255.0 207.95.6.101 1/1 1 S
54 209.159.40.0 255.255.255.0 207.95.6.101 1/1 1 S
55 209.159.41.0 255.255.255.0 207.95.6.101 1/1 1 S
56 209.159.42.0 255.255.255.0 207.95.6.101 1/1 1 S
57 209.159.43.0 255.255.255.0 207.95.6.101 1/1 1 S
58 209.159.44.0 255.255.255.0 207.95.6.101 1/1 1 S
59 209.159.45.0 255.255.255.0 207.95.6.101 1/1 1 S
60 209.159.46.0 255.255.255.0 207.95.6.101 1/1 1 S
```

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.

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.

```
BigIron RX# 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.

The following table lists the information displayed by the **show ip route** command.

TABLE 54 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.

TABLE 54 CLI display of IP route table (Continued)

This field...	Displays...
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 device. • 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 elapse time since an IP route was installed.

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.

```
BigIron RX# clear ip route
```

To clear route 209.157.22.0/24 from the IP routing table.

```
BigIron RX# 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 device, only those packets that are forwarded or generated by the CPU are included in the IP traffic statistics. Hardware forwarded packets are not included.

```
BigIron RX> sh ip traffic
IP Statistics
  146806 total received, 72952 mp received, 6715542 sent, 0 forwarded
  0 filtered, 0 fragmented, 0 bad header
  0 failed reassembly, 0 reassembled, 0 reassembly required
  0 no route, 0 unknown proto, 0 no buffer, 0 other errors, 0 rpf discard

ARP Statistics
  19022 total recv, 35761 req recv, 475 rep recv, 2803975 req sent, 1885 rep
sent
  0 pending drop, 0 invalid source, 0 invalid dest

ICMP Statistics
Received:
  9 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 8 echo, 1 echo reply
Sent:
  9 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect
  1 echo, 8 echo reply, 0 irdp advertisement, 0 irdp solicitation

UDP Statistics
  7230 received, 5604608 sent, 1020 no port, 0 input errors

TCP Statistics
  2706 in segments, 3689 out segments, 0 retransmission, 0 input errors
BigIron RX#
```

Syntax: show ip traffic

The **show ip traffic** command displays the following information.

TABLE 55 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.
no buffer	This information is used by *Brocade* customer support.
other errors	The number of packets that this device dropped due to error types other than the types listed above.

TABLE 55 CLI display of IP traffic statistics (Continued)

This field...	Displays...
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 *Brocade* 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 *Brocade* 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 *Brocade* 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 55 CLI display of IP traffic statistics (Continued)

This field...	Displays...
input errors	This information is used by *Brocade* 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.
RIP statistics The RIP statistics are derived from RFC 1058, "Routing Information Protocol".	
requests sent	The number of requests this device has sent to another RIP router for all or part of its RIP routing table.
requests received	The number of requests this device has received from another RIP router for all or part of this device's RIP routing table.
responses sent	The number of responses this device has sent to another RIP router's request for all or part of this device's RIP routing table.
responses received	The number of responses this device has received to requests for all or part of another RIP router's routing table.
unrecognized	This information is used by *Brocade* customer support.
bad version	The number of RIP packets dropped by the device because the RIP version was either invalid or is not supported by this device.
bad addr family	The number of RIP packets dropped because the value in the Address Family Identifier field of the packet's header was invalid.
bad req format	The number of RIP request packets this router dropped because the format was bad.
bad metrics	This information is used by *Brocade* customer support.
bad resp format	The number of responses to RIP request packets this router dropped because the format was bad.
resp not from rip port	This information is used by *Brocade* customer support.
resp from loopback	The number of RIP responses received from loopback interfaces.
packets rejected	This information is used by *Brocade* customer support.

Displaying TCP traffic statistics

You can use the show ip tcp traffic command to display TCP traffic statistics.

```
BigIron RX# show ip tcp traffic
TCP Statistics
233 active opens, 0 passive opens, 1659 failed attempts 117547
active resets, 0 passive resets, 116511 input errors 141627 in
segments, 18866 out segments, 71 retransmission
```

Syntax: show ip tcp traffic

This field...	Displays...
active opens	Number of TCP connection requests from the local router, resulting in outbound TCP SYNC packets
passive opens	Number of TCP connection requests from remote routers or hosts, resulting in outbound TCP SYNC-ACK packets
failed attempts	Number of unsuccessful TCP connection requests from either local or remote
active resets,	Number of TCP RESET packets sent by the local router
passive resets,	Number of normal TCP connections closed
input errors	Number of TCP packets received with error (header too short, checksum error, or not a listening TCP PORT)
in segments,	Number of TCP packet received
out segments,	Number of TCP packet sent
retransmission	Number of TCP packet re-transmitted

Chapter

Link Aggregation 8

In this chapter

Link aggregation overview

This chapter describes how to configure Link Aggregation Groups (LAG). Beginning with release 02.6.00 of the Multi-Service IronWare software, you can use a single interface to configure any of the following LAG types:

- **Static LAGs** – These trunk 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 switchswitch 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 device switches is established. In a keep alive LAG, LACP PDUs are exchanged between the 2 ports to determine if the connection between the switches is still active. If it is determined that the connection is no longer active, the ports are blocked.

NOTE
No trunk is created for “Keep Alive” LAGs

The new LAG configuration procedures supersede the previous configurations procedures for Trunks and Dynamic Link Aggregation. When a device switch is upgraded to release 02.6.00, any configurations for Trunks or Dynamic Link Aggregation defined in Multi-Service IronWare versions prior to 02.6.00 will be converted to a 02.6.00 (and later) compatible LAG configuration. Details about how this conversion is performed are described in “Migration from a pre-02.6.00 trunk or LACP configuration” on page 233 . The following are the major differences between in LAG configuration pre-02.6.00 and post-02.6.00 release:

- Beginning with release 02.6.00, a trunk is not created until a LAG is deployed using the **deploy** command.
- Beginning with release 02.6.00, LACP is not started until a dynamic LAG is deployed.

NOTE
Refer to the Dynamic Link Aggregation chapter in the *BigIron RX Series Configuration Guide* - Versions 02.5.00 and earlier for information on how to configure LACP in previous version of the Multi-Service IronWare software.

LAG formation rules

Given below are the LAG formation rules:

- 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 8 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.
- Trunk 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.
- Beginning with release 02.6.00, 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.
- VLAN and inner-VLAN translation.

  The trunk is rejected if any LAG port has VLAN or inner-VLAN translation configured
- Layer 2 requirements.

  The trunk is rejected if the trunk ports:

  - do not have the same untagged VLAN component.
  - do not share the same SuperSpan customer id (or cid).
  - do not share the same vlan membership
  - do not share the same uplink vlan membership
  - do not share the same protocol-vlan configuration
  - are configured as marble primary and secondary interfaces
- Layer 3 requirements.

  The trunk is rejected if any of the secondary trunk port has any Layer 3 configurations, such as Ipv4 or Ipv6 address, ospf, rip, ripng, isis, etc.
- Layer 4 (ACL) requirements.

  All trunk ports must have the same ACL configurations; otherwise, the trunk is rejected.
- The maximum number of ports supported in a LAG is 8.

  NOTE
  There are no SW limitations on the number of un-deployed or keep-alive LAGs. The trunk is a deployed on a static or dynamic LAG.
- Ports can be in only one LAG group. For example, port 1/4 cannot be in the LAG named “red” and in the LAG named “blue”.

- All the ports in a trunk group must be connected to the same device at the other end. For example, a if port 1/4 and 1/5 in Device 1 are in the same trunk group, both ports must be connected to a 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
  - QoS priority

  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.
- Make sure the device on the other end of the trunk link can support the same number of ports in the link.

Figure 13 displays and 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 13 Example of a 1-port keep alive LAG

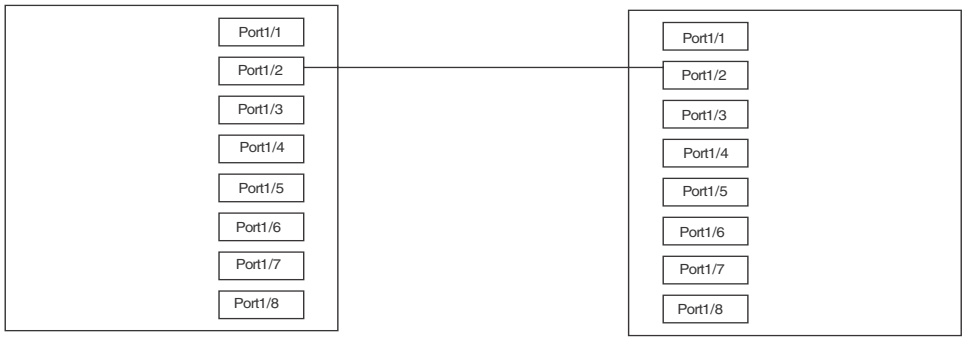


Figure 14 shows an example of a valid 2-port LAG link between devices where the ports on each end are on the same interface module. Ports in a valid 2-port LAG on one device are connected to two ports in a valid 2-port LAG on another device.

FIGURE 14 Example of 2-port LAG

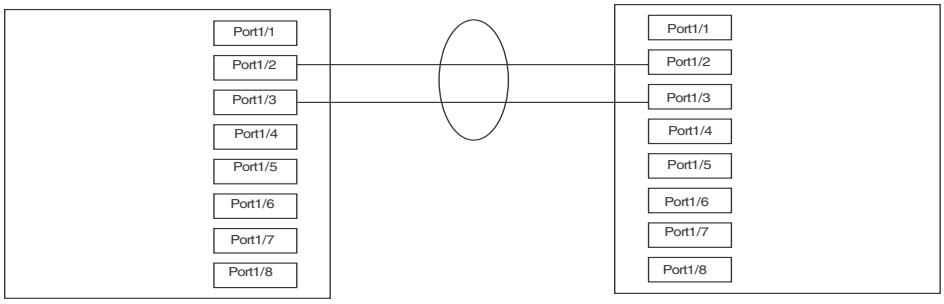
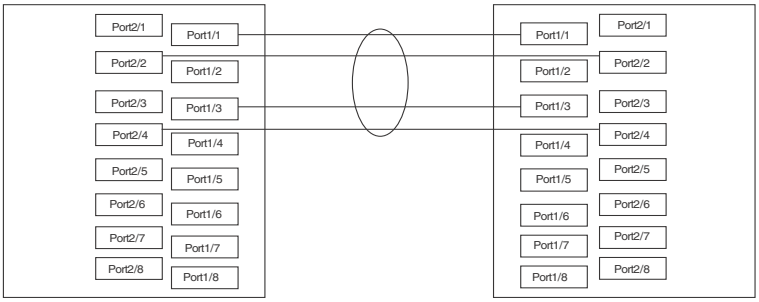


Figure 15 shows and example of two devices connected over a 4 port LAG where the ports on each end of the LAG are on different interface modules.

FIGURE 15 Examples of multi-slot, multi-port LAG



LAG load sharing

device switches can be configured for load sharing over a LAG by using the Hash Based Load Sharing method.

Hash based load sharing

The device switch 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 trunk index to identify them. Traffic from each flow is then distributed across the ports in the LAG group using a hash index as follows:

- For L2 traffic, the hash index is based on the following:
  - Layer-2 packets with an IPv4 payload: source IPv4 address, source mac addresses, destination IPv4 address, desination mac address and TCP/UDP source port and TCP/UDP destination port.
- For L3 traffic, the hash index is based on the following:
  - 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.
- For L2 VPN traffic, the hash index is based on the following:
  - Layer-2, non-IPv4.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.

The device switch uses the hash index in the following formula, using the modulo operator (written as "%" in C programming language).

(hash index)% (Number of trunk ports in a trunk group) = selected trunk port

Migration from a pre-02.6.00 trunk or LACP configuration

if you are upgrading from a version of the Multi-Service IronWare software prior to 02.6.00 and have either Trunks or LACP configured, the previous configuration will be automatically updated with the new commands to form an LAG that is equivalent to the previous configuration. To accomplish this, the old **trunk** and **link-aggregation** commands are maintained during startup configuration parsing, but disabled during normal configurations.

The following process is followed during the conversion of the **trunk** and **link-aggregation** to the new LAG commands.

1. For any static trunk configured using the **trunk ethernet <slot/port> to <slot/port>** command, the following conversion procedure is followed.
   a. A static LAG is created containing the port list specified in the **trunk** command. This LAG is then automatically deployed.
   b. The lowest-numbered port from the original trunk list is selected as the primary port of the LAG.
   c. Any **trunk config-trunk-ind** command, such as port name, is converted to the corresponding LAG commands.
   d. The default load balancing scheme previously known as **server** is converted to the default **hash-based** load balancing scheme.
   e. The converted LAG is named "LAG_x", where "x" is a unique number assigned by the system starting from 1.
2. For any dynamic link aggregation (LACP) group configured using the port-level **link-aggregate** commands, the following conversion procedure is followed.

a. A dynamic LAG is created by grouping all ports in the original configuration having the same link-aggregation key.

b. If **link-aggregate active/passive** is configured originally, the converted dynamic LAG will be configured as deployed. Otherwise it will not be converted because such ports were originally not operating under LACP.

c. If the original mode is passive, the converted dynamic LAG will be configured as **deploy passive**. Otherwise active mode is the default.

d. The timeout configuration set by the command **link-aggregate configure timeout** will be converted to the **lacp-timeout** command.

e. Individual port priority set by the command **link-aggregate configure port-priority** will be converted to the **lacp-port-priority** command.

f. While the value of the **link-aggregate configure key** command is used in the conversion in determining the set of ports that form an LAG, in the new LAG user interface, there is no need for a user to explicitly configure a key. Each dynamic LAG will automatically select a unique key for the system. Hence the original configured key will not be retained.

g. The command **link-aggregate configure system-priority** is retired and will not be directly converted. This value is currently not in use by the system's LACP protocol processing, and will maintain a default value of 1.

h. The lowest-numbered port will be selected as the primary port of the LAG.

i. The load balancing scheme in the command **link-aggregate configure type** is automatically converted to the default **hash-based** load balancing scheme.

j. Port names configured in the original interface configuration will be converted to port names within the LAG.

k. The converted LAG will be named "LAG_x", where “x” is a unique number assigned by the system starting from 1.

Configuration of 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 8 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.
- **Specifying the Trunk Threshold for a Trunk 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)

Before setting-up ports or configuring any other aspects of a LAG, you must create it as shown in the following.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)#
```

Syntax: [no] lag <*lag-name*> static | dynamic | keep-alive

Refer to "Allowable characters for LAG names" on page 14 for guidelines on LAG naming conventions.

The **static** option specifies that the LAG with the name specified by the <*lag-name*> variable will be configured as a static LAG. The static LAG configuration is much the same as the Trunk feature available in releases prior to 02.6.00.

The **dynamic** option specifies that the LAG with the name specified by the <lag-name> variable will be configured as a dynamic LAG. The dynamic LAG configuration is much the same as the LACP feature available in releases prior to 02.6.00.

The **keep-alive** option specifies that the LAG with the name specified by the <lag-name> variable will be configured as a keep-alive LAG. The keep-alive LAG configuration is a new configuration option to configure a LAG for use in keep alive applications similar to the UDLD feature.

Adding ports to a LAG

A static or dynamic LAG can consist of from 2 to 20 ports of the same type and speed that are on any interface module within the device chassis. A keep alive LAG consists of only one port.

To configure the static LAG named "blue" with two ports, use the following command.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# ports ethernet 3/1 ethernet 7/2
```

Syntax: [no] ports ethernet <*slot/port*> [to <*slot/port*>] [ethernet <*slot/port*>]

The ports added to a LAG are **ethernet** as specified for the **slot/port** where they reside. The ports can be added to the LAG sequentially as shown in the following example.

```
BigIron RX(config-lag-blue)# ports ethernet 3/1 ethernet 7/2 ethernet 4/3 ethernet
3/
```

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.

```
BigIron RX(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.

```
BigIron RX(config-lag-blue)# ports ethernet 3/1 to 3/4 ethernet 10/2
```

NOTE
Beginning with release 02.6.00, a port can be added to or deleted from a LAG only if the LAG is not currently deployed.

Configuring the primary port for a LAG

In previous versions of the Multi-Service IronWare software, the lowest number port was assigned as the primary port in a trunk or LACP configuration. In version 02.6.00 and later, 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.

```
BigIron RX(config)# lag blue static
BigIron RX(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.

Specifying the trunk threshold for a trunk Group

You can configure the device switch to disable all of the ports in a trunk group when the number of active member ports drops below a specified threshold value. For example, if a trunk group has 8 ports, and the threshold for the trunk group is 5, then the trunk group is disabled if the number of available ports in the trunk group drops below 5. If the trunk group is disabled, then traffic is forwarded over a different link or trunk group.

NOTE
This configuration is only applicable for configuration of a static or dynamic LAGs.

For example, the following commands establish a trunk group consisting of 4 ports, then establish a threshold for this trunk group of 3 ports.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# ports ethernet 3/1 to 3/4
BigIron RX(config-lag-blue)# trunk-threshold 3
```

In this example, if the number of active ports drops below 3, then all the ports in the trunk group are disabled.

Syntax: trunk-threshold <*number*>

You can specify a threshold from 1 (the default) up to the number of ports in the trunk 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
Trunk threshold 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 trunk will not function properly.

Configuring LACP port priority

In a dynamic or keep alive LAG, the port priority determines the active and standby links. The other ports (with lower priorities) become standby ports in the trunk group.

```
BigIron RX(config)# lag blue dynamic
BigIron RX(config-lag-blue)# lacp-port-priority 100000
```

Syntax: [no] lacp-port-priority <*slot/port*> <*number*>

For a port specified by the <*slot/port*> variable, you can specify a priority in the <*number*> variable from 0 – 65535. A higher value indicates a lower priority. The default is 1.

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 or long. Once a port is configured with a timeout option, it will remain in that timeout mode whether it's up or down, or part of a trunk or not.

All the ports in a trunk should have the same timeout mode. This is checked when the LAG is enabled on ports. To configure a port for a short LACP timeout use the following command.

```
BigIron RX(config)# lag blue dynamic
BigIron RX(config-lag-blue)# lacp-timeout short
```

Syntax: [no] lacp-timeout [long | short]

The **long** parameter configures the port for the long timeout mode.

The **short** parameter configures the port for the short timeout mode.

NOTE
This configuration is only applicable for configuration of a dynamic or keep-alive LAGs.

Deploying a LAG

After configuring a LAG, you must explicitly enable it before it takes begins aggregating traffic. This is accomplished using the **deploy** command within the LAG configuration. Once the **deploy** command is executed, 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. Once a non keep-alive LAG is deployed, a trunk is formed. If there is only one port in the LAG, a single port trunk is formed. For a dynamic LAG, LACP is stared for each LAG port. For a keep-alive LAG, no trunk is formed and LACP is started on the LAG port.

You can deploy a LAG as shown in the following for the “blue” LAG.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# deploy
```

Syntax: [no] deploy [forced | passive]

When the **deploy** command is executed:

For a static and dynamic LAGs, the current trunk veto mechanism is invoked to make sure the trunk can be formed. If the trunk is not vetoed, a trunk 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 trunk 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 trunk 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

ACL-based mirroring can be configured for an individual port within a LAG using the **acl-mirror-port** command, as shown in the following.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# deploy
BigIron RX(config-lag-blue)# acl-mirror-port ethe-port-monitored 3/1
```

Syntax: [no] acl-mirror-port ethe-port-monitored [slot/port] | named-port-monitored [name]

Use the **ethe-port-monitored** option with the appropriate **[slot/port]** variable to specify a Ethernet port that you want to provide ACL mirroring for.

Use the **named-port-monitored** option with the appropriate **[slot/port]** variable to specify a named port that you want to provide ACL mirroring for.

NOTE
Mirror (analyzer) ports cannot be assigned to the 16x10G card. You can monitor traffic on 16x10 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.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# deploy
BigIron RX(config-lag-blue)# disable ethernet 3/1
```

Syntax: [no] disable ethernet [slot/port] | named [name]

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.

Enabling ports within a LAG

You can enable an individual port within a trunk using the disable command within the LAG configuration as shown in the following.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# deploy
BigIron RX(config-lag-blue)# enable ethernet 3/1
```

Syntax: [no] enable ethernet [slot/port] | named [name]

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.

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, or Named ports. You can monitor the primary port or a secondary port individually.

NOTE
You can use only one mirror port for each monitored trunk port. To monitor traffic on an individual port in a trunk group, enter commands such as the following:

This command enables monitoring of an individual port within a LAG.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# deploy
BigIron RX(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] 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.

The **ethernet <slot/port>** parameter specifies the port to which the traffic analyzer is attached.

The **input | output | both** parameters specify the traffic direction to be monitored.

NOTE
Mirror (analyzer) ports cannot be assigned to the 16x10G card. You can monitor traffic on 16x10 ports.

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.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# deploy
BigIron RX(config-lag-blue)# port-name orange ethernet 3/1
```

Syntax: [no] port-name <*text*> ethernet [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.

Refer to "Allowable characters for LAG names" on page 14 for guidelines on LAG naming conventions.

Enabling sFlow forwarding on a port within 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.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# deploy
BigIron RX(config-lag-blue)# sflow-forwarding ethernet 3/1
```

Syntax: [no] sflow-forwarding ethernet [slot/port] | port-name [text]

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.

Setting the sFlow sampling rate for a port within 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.

```
BigIron RX(config)# lag blue static
BigIron RX(config-lag-blue)# deploy
BigIron RX(config-lag-blue)# sflow-subsampling ethernet 3/1 512
```

Syntax: [no] sflow-subsampling ethernet [slot/port] | port-name [text] <*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.

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.

Displaying LAG information

You can display LAG information for a device switch in either a **full** or **brief** mode. The examples below show both options of the **show lag** command.

```
BigIron RX# show lag brief
Total number of LAGs:          4
Total number of deployed LAGs: 3
Total number of trunks created:3 (31 available)
LACP System Priority / ID:     0001 / 0004.80a0.4000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3

LAG           Type   Deploy Trunk 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      ethe 4/1 ethe 4/3 ethe 4/5
s1            static   N    none  32/3     ethe 13/4 ethe 32/3 to 32/4

BigIron RX# show lag
Total number of LAGs:          4
Total number of deployed LAGs: 3
Total number of trunks created:3 (31 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
   LACP Key:      104

Deployment:  Trunk ID 3

Port  Link L2 State  Dupl Speed Trunk 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
   LACP Key:      105
```

```
Deployment:  Trunk ID 1

Port  Link L2 State  Dupl Speed Trunk 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*> [brief] [deployed] [dynamic] [keep-alive] [static]

Table 56 describes the information displayed by the **show lag** command.

TABLE 56 Show LAG information

This field...	Displays...
Total number of LAGS	The total number of LAGs that have been configured on the switch.
Total number of Deployed LAGS	The total number of LAGs on the switch that are currently deployed.
Total number of Trunks Created	The total number of trunks that have been created on the LAG. The total number of Trunks available are shown also. Since keep-alive LAGs do not use a trunk ID, they are not listed and do not subtract for the number of trunks available.
LACP System Priority /ID	The system priority configured for the switch . The ID is the system priority which is the base MAC address of the switch.
LACP Long timeout	
LACP Short timeout	
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.
Trunk	The trunk 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.
LACP Key	The link aggegation key for the LAG.
Deployment	The Trunk ID number.
Port	The chassis slot and port number of the interface.

TABLE 56 Show LAG information (Continued)

This field...	Displays...
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
Speed	The bandwidth of the interface.
Trunk	The Trunk 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 trunk 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.

TABLE 56 Show LAG information (Continued)

This field...	Displays...
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 trunk 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 trunk 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 trunk link: • **Col** – The port is ready to send traffic over the trunk link. • **No** – The port is not ready to send traffic over the trunk link.
Dis	Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the trunk link: • **Dis** – The port is ready to receive traffic over the trunk link. • **No** – The port is not ready to receive traffic over the trunk link.
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 trunk group. An LACP port is blocked until it becomes part of a trunk. 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 device switch 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.

```
BigIron RX# 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

BigIron RX# 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*>]

Chapter

Configuring LLDP

9

In this chapter

Terms used in this chapter

Link Layer Discovery Protocol (LLDP) – The Layer 2 network discovery protocol described in the IEEE 802.1AB standard, *Station and Media Access Control Connectivity Discovery*. This protocol enables a station to advertise its capabilities to, and to discover, other LLDP-enabled stations in the same 802 LAN segments.

LLDP Agent – The protocol entity that implements LLDP for a particular IEEE 802 device. Depending on the configured LLDP operating mode, an LLDP agent can send and receive LLDP advertisements (frames), or send LLDP advertisements only, or receive LLDP advertisements only.

LLDPDU (LLDP Data Unit) – A unit of information in an LLDP packet that consists of a sequence of short variable length information elements, known as **TLVs**.

MIB (Management Information Base) – A virtual database that identifies each manageable object by its name, syntax, accessibility, and status, along with a text description and unique object identifier (OID). The database is accessible by a Network Management Station (NMS) using a management protocol such as the Simple Network Management Protocol (SNMP).

Network Connectivity Device – A forwarding 802 LAN device, such as a router, switch, or wireless access point.

Station – A node in a network.

TLV (Type-Length-Value) – An information element in an LLDPDU that describes the type of information being sent, the length of the information string, and the value (actual information) that will be transmitted.

TTL (Time-to-Live) – Specifies the length of time that the receiving device should maintain the information acquired through LLDP in its MIB.

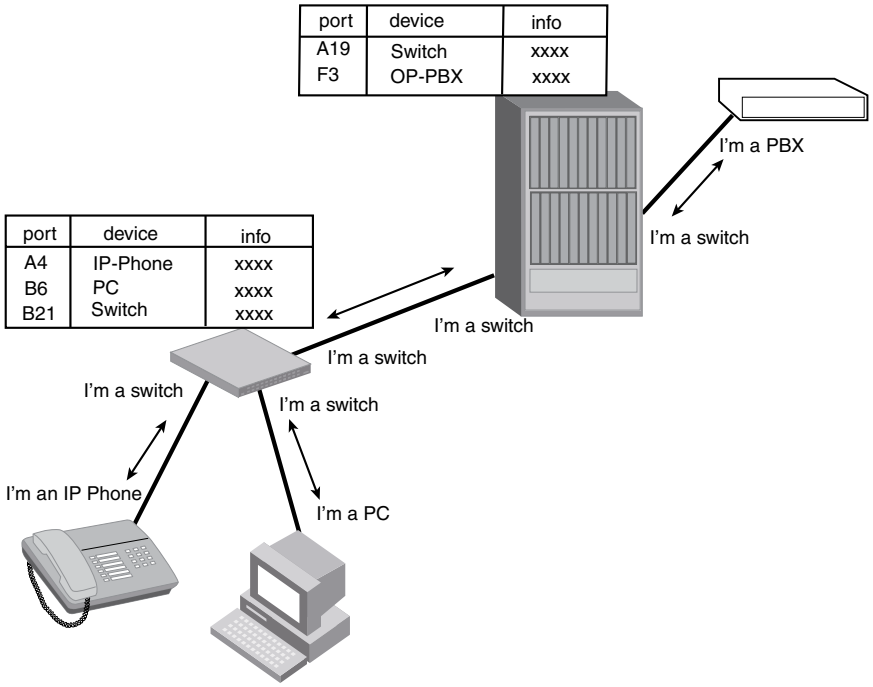
LLDP overview

LLDP enables a station attached to an IEEE 802 LAN or MAN to advertise its capabilities to, and to discover, other stations in the same 802 LAN segments.

The information distributed through LLDP (the advertisement) is stored by the receiving device in a standard Management Information Base (MIB), accessible by a Network Management System (NMS) using a management protocol such as the Simple Network Management Protocol (SNMP). The information also can be viewed through the CLI, using **show LLDP** commands.

Figure 16 illustrates LLDP connectivity.

FIGURE 16 LLDP Connectivity



Benefits of LLDP

LLDP provides the following benefits:

- Network Management:
  - Simplifies the use of and enhances the ability of network management tools in multi-vendor environments
  - Enables discovery of accurate physical network topologies such as which devices are neighbors and through which ports they connect
  - Enables discovery of stations in multi-vendor environments
- Network Inventory Data:
  - Supports optional system name, system description, system capabilities and management address

  - System description can contain the device's product name or model number, version of hardware type, and operating system
  - Provides device capability, such as switch, router, or WLAN access port
- Network troubleshooting:
  - Information generated through LLDP can be used to detect speed and duplex mismatches
  - Accurate topologies simplify troubleshooting within enterprise networks
  - Can discover devices with misconfigured or unreachable IP addresses

General operating principles

LLDP use the services of the Data Link sublayers, Logical Link Control and Media Access Control, to transmit and receive information to and from other ***LLDP Agents*** (protocol entities that implement LLDP).

LLDP is a one-way protocol. An LLDP agent can transmit and receive information to and from another LLDP agent located on an adjacent device, but it cannot solicit information from another LLDP agent, nor can it acknowledge information received from another LLDP agent.

Operating modes

When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. You can disable a port's ability to transmit and receive LLDP packets, or change the operating mode to one of the following:

- Transmit LLDP information only
- Receive LLDP information only

Transmit mode

An LLDP agent sends LLDP packets to adjacent LLDP-enabled devices. The LLDP packets contain information about the transmitting device and port.

An LLDP agent initiates the transmission of LLDP packets whenever the transmit countdown timing counter expires, or whenever LLDP information has changed. When a transmit cycle is initiated, the LLDP manager extracts the MIB objects and formats this information into TLVs. The TLVs are inserted into an LLDPDU, addressing parameters are prepended to the LLDPDU, and the information is sent out LLDP-enabled ports to adjacent LLDP-enabled devices.

Receive mode

An LLDP agent receives LLDP packets from adjacent LLDP-enabled devices. The LLDP packets contain information about the transmitting device and port.

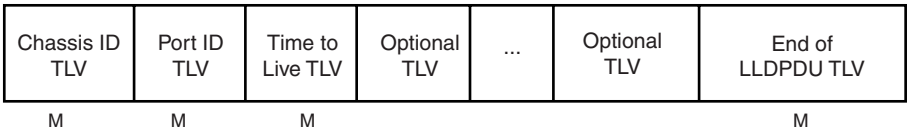
When an LLDP agent receives LLDP packets, it checks to ensure that the LLDPDUs contain the correct sequence of mandatory TLVs, then validates optional TLVs. If the LLDP agent detects any errors in the LLDPDUs and TLVs, it drops them in software. TLVs that are not recognized but do not contain basic formatting errors, are assumed to be valid and are assigned a temporary identification index and stored for future possible alter retrieval by network management. All validated TLVs are stored in the neighbor database.

LLDP packets

LLDP agents transmit information about a sending device or port in packets called LLDP Data Units (LLDPDUs). All the LLDP information to be communicated by a device is contained within a single 1500 byte packet. A device receiving LLDP packets is not permitted to combine information from multiple packets.

As shown in Figure 17, each LLDPDU has three mandatory TLVs, an End of LLDPDU TLV, plus optional TLVs as selected by network management.

FIGURE 17 LLDPDU packet format



Each LLDPDU consists of an untagged Ethernet header and a sequence of short, variable length information elements known as TLVs.

TLVs have Type, Length, and Value fields, where:

- **Type** identifies the kind of information being sent
- **Length** indicates the length (in octets) of the information string
- **Value** is the actual information being sent (for example, a binary bit map or an alpha-numeric string containing one or more fields).

TLV support

This section lists the LLDP TLV support.

LLDP TLVs

There are two types of LLDP TLVs, as specified in the IEEE 802.3AB standard:

- **Basic Management TLVs** consist of both optional general system information TLVs as well as mandatory TLVs.

  Mandatory TLVs cannot be manually configured. They are always the first three TLVs in the LLDPDU, and are part of the packet header.

  General system information TLVs are optional in LLDP implementations and are defined by the Network Administrator.

  Brocade devices support the following Basic Management TLVs:

  - Chassis ID (mandatory)
  - Port ID (mandatory)
  - Time to Live (mandatory)
  - Port description
  - System name
  - System description

- System capabilities
- Management address
- End of LLDPDU

- **Organizationally-specific TLVs** are optional in LLDP implementations and are defined and encoded by individual organizations or vendors. These TLVs include support for, but are not limited to, the IEEE 802.1 and 802.3 standards and the TIA-1057 standard.

  Brocade devices support the following Organizationally-specific TLVs:

  - **802.1 organizationally-specific TLVs**

    Port VLAN ID

    VLAN name TLV

  - **802.3 organizationally-specific TLVs**

    MAC/PHY configuration/status

    Link aggregation

    Maximum frame size

Mandatory TLVs

When an LLDP agent transmits LLDP packets to other agents in the same 802 LAN segments, the following mandatory TLVs are always included:

- Chassis ID
- Port ID
- Time to Live (TTL)

This section describes the above TLVs in detail.

Chassis ID

The Chassis ID identifies the device that sent the LLDP packets.

There are several ways in which a device may be identified. A chassis ID subtype, included in the TLV and shown in Table 57, indicates how the device is being referenced in the Chassis ID field.

TABLE 57 Chassis ID subtypes

ID Subtype	Description
0	Reserved
1	Chassis component
2	Interface alias
3	Port component
4	MAC address
5	Network address
6	Interface name
7	Locally assigned
8 – 255	Reserved

Brocade devices use chassis ID subtype 4, the base MAC address of the device. Other third party devices may use a chassis ID subtype other than 4. The chassis ID will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Chassis ID (MAC address):  0012.f233.e2c0
```

The chassis ID TLV is always the first TLV in the LLDPDU.

Port ID

The Port ID identifies the port from which LLDP packets were sent.

There are several ways in which a port may be identified, as shown in Table 58. A port ID subtype, included in the TLV, indicates how the port is being referenced in the Port ID field.

TABLE 58 Port ID subtypes

ID Subtype	Description
0	Reserved
1	Interface alias
2	Port component
3	MAC address
4	Network address
5	Interface name
6	Agent circuit ID
7	Locally assigned
8 – 255	Reserved

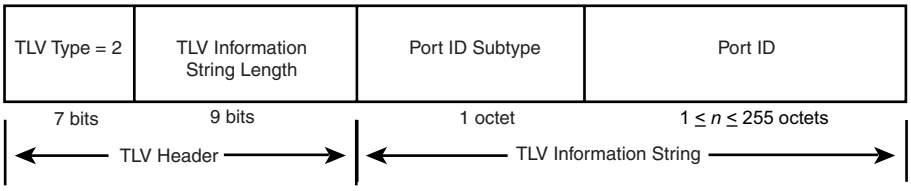
Brocade devices use port ID subtype 3, the permanent MAC address associated with the port. Other third party devices may use a port ID subtype other than 3. The port ID appears similar to the following on the remote device, and in the CLI display output on the Brocade device (show lldp local-info).

```
Port ID (MAC address):  0012.f233.e2d3
```

The LLDPDU format is shown in "LLDPDU packet format" on page 250.

The Port ID TLV format is shown below.

FIGURE 18 Port ID TLV packet format



TTL value

The Time to Live (TTL) Value is the length of time the receiving device should maintain the information acquired through LLDP in its MIB.

The TTL value is automatically computed based on the LLDP configuration settings. The TTL value will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (show lldp local-info).

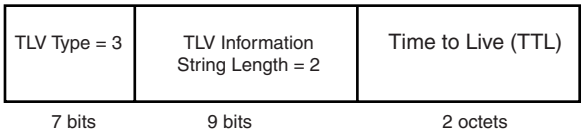
```
Time to live: 40 seconds
```

- If the TTL field has a value other than zero, the receiving LLDP agent is notified to completely replace all information associated with the LLDP agent or port with the information in the received LLDPDU.
- If the TTL field value is zero, the receiving LLDP agent is notified that all system information associated with the LLDP agent or port is to be deleted. This TLV may be used, for example, to signal that the sending port has initiated a port shutdown procedure.

The LLDPDU format is shown in "LLDPDU packet format" on page 250.

The TTL TLV format is shown below.

FIGURE 19 TTL TLV packet format



MIB support

Brocade devices support the following standard MIB modules:

- LLDP-MIB
- LLDP-EXT-DOT1-MIB
- LLDP-EXT-DOT3-MIB

Syslog messages

Syslog messages for LLDP provide management applications with information related to MIB data consistency and general status. These Syslog messages correspond to the lldpRemTablesChange SNMP notifications.

Configuring LLDP

This section describes how to enable and configure LLDP.

Table 59 lists the LLDP global-level tasks and the default behavior/value for each task.

TABLE 59 LLDP global configuration tasks and default behavior / value

Global task	Default behavior / value when LLDP is enabled
Enabling LLDP on a global basis	Disabled
Specifying the maximum number of LLDP neighbors per device	Automatically set to 392 neighbors per device

TABLE 59 LLDP global configuration tasks and default behavior / value (Continued)

Global task	Default behavior / value when LLDP is enabled
Specifying the maximum number of LLDP neighbors per port	Automatically set to 4 neighbors per port
Enabling SNMP notifications and Syslog messages	Disabled
Changing the minimum time between SNMP traps and Syslog messages	Automatically set to 2 seconds when SNMP notifications and Syslog messages for LLDP are enabled
Enabling and disabling TLV advertisements	When LLDP transmit is enabled, by default, the Brocade device will automatically advertise LLDP capabilities, except for the system description, VLAN name, and power-through-MDI information, which may be configured by the system administrator. Also, if desired, you can disable the advertisement of individual TLVs.
Changing the minimum time between LLDP transmissions	Automatically set to 2 seconds
Changing the interval between regular LLDP transmissions	Automatically set to 30 seconds
Changing the holdtime multiplier for transmit TTL	Automatically set to 4
Changing the minimum time between port reinitializations	Automatically set to 2 seconds

Configuration notes and considerations

- LLDP is supported on Ethernet interfaces only.
- If a port is 802.1X-enabled, the transmission and reception of LLDP packets will only take place while the port is authorized.
- Cisco Discovery Protocol (CDP) and Foundry Discovery Protocol (FDP) run independently of LLDP. Therefore, these discovery protocols can run simultaneously on the same device.
- By default, the Brocade device limits the number of neighbors per port to four, and staggers the transmission of LLDP packets on different ports, in order to minimize any high-usage spikes to the CPU.
- By default, the Brocade device forwards
- Ports that are in blocking mode (spanning tree) can still receive LLDP packets from a forwarding port.
- Auto-negotiation status indicates what is being advertised by the port for 802.3 auto-negotiation.

Enabling and disabling LLDP

LLDP is enabled by default on individual ports. However, to run LLDP, you must first enable it on a global basis (on the entire device).

To enable LLDP globally, enter the following command at the global CONFIG level of the CLI.

```
FastIron(config)#lldp run
```

Syntax: [no] lldp run

Changing a port's LLDP operating mode

LLDP packets are not exchanged until LLDP is enabled on a global basis. When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. You can disable a port's ability to transmit and receive LLDP packets, or change the operating mode to one of the following:

- Transmit LLDP information only
- Receive LLDP information only

You can configure a different operating mode for each port on the Brocade device. For example, you could disable the receipt and transmission of LLDP packets on port e 2/1, configure port e 2/3 to only receive LLDP packets, and configure port e 2/5 to only transmit LLDP packets.

The following sections show how to change the operating mode.

Enabling and disabling receive and transmit mode

To disable the receipt and transmission of LLDP packets on individual ports, enter a command such as the following at the Global CONFIG level of the CLI.

```
FastIron(config)#no lldp enable ports e 2/4 e 2/5
```

The above command disables LLDP on ports 2/4 and 2/5. These ports will not transmit nor receive LLDP packets.

To enable LLDP on a port after it has been disabled, enter the following command.

```
FastIron(config)#lldp enable ports e 2/4
```

Syntax: [no] lldp enable ports ethernet <slotnum/portnum> | all

Use the [no] form of the command to disable the receipt and transmission of LLDP packets on a port.

You can list all of the ports individually, use the keyword to to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword all instead of listing the ports individually.

Enabling and disabling receive only mode

When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. To change the LLDP operating mode from receive and transmit mode to receive only mode, simply disable the transmit mode. Enter a command such as the following at the Global CONFIG level of the CLI.

```
FastIron(config)#no lldp enable transmit ports e 2/4 e 2/5 e 2/6
```

The above command changes the LLDP operating mode on ports 2/4, 2/5, and 2/6 from transmit and receive mode to receive only mode.

To change a port's LLDP operating mode from transmit only to receive only, first disable the transmit only mode, then enable the receive only mode. Enter commands such as the following.

```
FastIron(config)#no lldp enable transmit ports e 2/7 e 2/8 e 2/9
FastIron(config)#lldp enable receive ports e 2/7 e 2/8 e 2/9
```

The above commands change the LLDP operating mode on ports 2/7, 2/8, and 2/9, from transmit only to receive only. Note that if you do not disable the transmit only mode, you will configure the port to both transmit and receive LLDP packets.

Syntax: [no] lldp enable receive ports ethernet <*slotnum/portnum*> | all

Use the **[no]** form of the command to disable the receive only mode.

You can list all of the ports individually, use the keyword to to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword all instead of listing the ports individually.

Enabling and disabling transmit only mode

When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. To change the LLDP operating mode to transmit only mode, simply disable the receive mode. Enter a command such as the following at the Global CONFIG level of the CLI.

```
FastIron(config)#no lldp enable receive ports e 2/4 e 2/5 e 2/6
```

The above command changes the LLDP operating mode on ports 2/4, 2/5, and 2/6 from transmit and receive mode to transmit only mode. Any incoming LLDP packets will be dropped in software.

To change a port's LLDP operating mode from receive only to transmit only, first disable the receive only mode, then enable the transmit only mode. For example, enter commands such as the following at the Global CONFIG level of the CLI.

```
FastIron(config)#no lldp enable receive ports e 2/7 e 2/8
FastIron(config)#lldp enable transmit ports e 2/7 e 2/8
```

The above commands change the LLDP operating mode on ports 2/7 and 2/8 from receive only mode to transmit only mode. Any incoming LLDP packets will be dropped in software. Note that if you do not disable receive only mode, you will configure the port to both receive and transmit LLDP packets.

Syntax: **[no]** lldp enable transmit ports ethernet <*slotnum/portnum*> | all

Use the **[no]** form of the command to disable the *transmit only* mode.

Use the **[no]** form of the command to disable the *receive only* mode.

You can list all of the ports individually, use the keyword to to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword all instead of listing the ports individually.

Specifying the maximum number of LLDP neighbors

You can change the limit of the number of LLDP neighbors for which LLDP data will be retained, per device as well as per port.

Per device

You can change the maximum number of neighbors for which LLDP data will be retained for the entire system.

For example, to change the maximum number of LLDP neighbors for the entire device to 26, enter the following command.

```
FastIron(config)#lldp max-total-neighbors 26
```

Syntax: [no] lldp max-total-neighbors <*value*>

Use the **[no]** form of the command to remove the static configuration and revert to the default value of 392.

where <*value*> is a number between 16 and 65536. The default number of LLDP neighbors per device is 392.

Use the **show lldp** command to view the configuration.

Per port

You can change the maximum number of LLDP neighbors for which LLDP data will be retained for each port. By default, the maximum number is four and you can change this to a value between one and 64.

For example, to change the maximum number of LLDP neighbors to six, enter the following command.

```
FastIron(config)#lldp max-neighbors-per-port 6
```

Syntax: [no] lldp max-neighbors-per-port <*value*>

Use the **[no]** form of the command to remove the static configuration and revert to the default value of four.

where <*value*> is a number from 1 to 64. The default is number of LLDP neighbors per port is four.

Use the **show lldp** command to view the configuration.

Enabling LLDP SNMP notifications and Syslog messages

SNMP notifications and Syslog messages for LLDP provide management applications with information related to MIB data updates and general status.

When you enable LLDP SNMP notifications, corresponding Syslog messages are enabled as well. When you enable LLDP SNMP notifications, the device will send traps and corresponding Syslog messages whenever there are changes to the LLDP data received from neighboring devices.

LLDP SNMP notifications and corresponding Syslog messages are disabled by default. To enable them, enter a command such as the following at the Global CONFIG level of the CLI.

```
BigIron RX(config)#lldp enable snmp notifications ports e 4/2 to 4/6
```

The above command enables SNMP notifications and corresponding Syslog messages on ports 4/2 and 4/6. By default, the device will send no more than one SNMP notification and Syslog message within a five second period. If desired, you can change this interval.

Syntax: [no] lldp enable snmp notifications ports ethernet <*slotnum/portnum*> | all

You can list all of the ports individually, use the keyword to to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword all instead of listing the ports individually.

Specifying the minimum time between SNMP traps and Syslog messages

When SNMP notifications and Syslog messages for LLDP are enabled, the device will send no more than one SNMP notification and corresponding Syslog message within a five second period. If desired, you can throttle the amount of time between transmission of SNMP traps (lldpRemTablesChange) and Syslog messages from five seconds up to a value equal to one hour (3600 seconds).

NOTE
Because LLDP Syslog messages are rate limited, some LLDP information given by the system will not match the current LLDP statistics (as shown in the **show lldp statistics** command output).

To change the minimum time interval between traps and Syslog messages, enter a command such as the following.

```
FastIron(config)#lldp snmp-notification-interval 60
```

When the above command is applied, the LLDP agent will send no more than one SNMP notification and Syslog message every 60 seconds.

Syntax: [no] lldp snmp-notification-interval <*seconds*>

where <*seconds*> is a value between 5 and 3600. The default is 5 seconds.

Changing the minimum time between LLDP transmissions

The LLDP transmit delay timer limits the number of LLDP frames an LLDP agent can send within a specified time frame. When you enable LLDP, the system automatically sets the LLDP transmit delay timer to two seconds. If desired, you can change the default behavior from two seconds to a value between 1 and 8192 seconds.

NOTE
The LLDP transmit delay timer must not be greater than one quarter of the LLDP transmission interval (CLI command **lldp transmit-interval**).

The LLDP transmit delay timer prevents an LLDP agent from transmitting a series of successive LLDP frames during a short time period, when rapid changes occur in LLDP. It also increases the probability that multiple changes, rather than single changes, will be reported in each LLDP frame.

To change the LLDP transmit delay timer, enter a command such as the following at the Global CONFIG level of the CLI.

```
FastIron(config)#lldp transmit-delay 7
```

The above command causes the LLDP agent to wait a minimum of seven seconds after transmitting an LLDP frame and before sending another LLDP frame.

Syntax: [no] lldp transmit-delay <*seconds*>.

where <*seconds*> is a value between 1 and 8192. The default is two seconds. Note that this value must not be greater than one quarter of the LLDP transmission interval (CLI command **lldp transmit-interval**).

Changing the interval between regular LLDP transmissions

The LLDP transmit interval specifies the number of seconds between regular LLDP packet transmissions. When you enable LLDP, by default, the device will wait 30 seconds between regular LLDP packet transmissions. If desired, you can change the default behavior from 30 seconds to a value between 5 and 32768 seconds.

To change the LLDP transmission interval, enter a command such as the following at the Global CONFIG level of the CLI.

```
FastIron(config)#lldp transmit-interval 40
```

The above command causes the LLDP agent to transmit LLDP frames every 40 seconds.

Syntax: [no] lldp transmit-interval <*seconds*>

where <*seconds*> is a value from 5 to 32768. The default is 30 seconds.

NOTE
Setting the transmit interval or transmit holdtime multiplier to inappropriate values can cause the LLDP agent to transmit LLDPDUs with TTL values that are excessively high. This in turn can affect how long a receiving device will retain the information if it is not refreshed.

Changing the holdtime multiplier for transmit TTL

The holdtime multiplier for transmit TTL is used to compute the actual time-to-live (TTL) value used in an LLDP frame. The TTL value is the length of time the receiving device should maintain the information in its MIB. When you enable LLDP, the device automatically sets the holdtime multiplier for TTL to four. If desired, you can change the default behavior from four to a value between two and ten.

To compute the TTL value, the system multiplies the LLDP transmit interval by the holdtime multiplier. For example, if the LLDP transmit interval is 30 and the holdtime multiplier for TTL is 4, then the value 120 is encoded in the TTL field in the LLDP header.

To change the holdtime multiplier, enter a command such as the following at the Global CONFIG level of the CLI.

```
FastIron(config)#lldp transmit-hold 6
```

Syntax: [no] lldp transmit-hold <*value*>.

where <*value*> is a number from 2 to 10. The default value is 4.

NOTE
Setting the transmit interval or transmit holdtime multiplier to inappropriate values can cause the LLDP agent to transmit LLDPDUs with TTL values that are excessively high. This in turn can affect how long a receiving device will retain the information if it is not refreshed.

Changing the minimum time between port reinitializations

The LLDP re-initialization delay timer specifies the minimum number of seconds the device will wait from when LLDP is disabled on a port, until it will honor a request to re-enable LLDP on that port. When you enable LLDP, the system sets the re-initialization delay timer to two seconds. If desired, you can change the default behavior from two seconds to a value between one and ten seconds.

To set the re-initialization delay timer, enter a command such as the following at the Global CONFIG level of the CLI.

```
FastIron(config)#lldp reinit-delay 5
```

The above command causes the device to wait five seconds after LLDP is disabled, before attempting to honor a request to re-enable it.

Syntax: [no] lldp reinit-delay <*seconds*>

where <*seconds*> is a value from 1 – 10. The default is two seconds.

LLDP TLVs advertised by the Brocade device

When LLDP is enabled on a global basis, the Brocade device will automatically advertise the following information, except for the features noted:

General system information:

- Management address
- Port description
- System capabilities
- System description (not automatically advertised)
- System name

802.1 capabilities:

- VLAN name (not automatically advertised)
- Untagged VLAN ID

802.3 capabilities:

- Link aggregation information
- MAC/PHY configuration and status
- Maximum frame size

The above TLVs are described in detail in the following sections.

NOTE
The system description, VLAN name, and power-through-MDI information TLVs are not automatically enabled. The following sections show how to enable these advertisements.

General system information

Except for the system description, the Brocade device will advertise the following system information when LLDP is enabled on a global basis:

- Management address
- Port description
- System capabilities
- System description (not automatically advertised)
- System name

Management address

The management address is an IPv4 address that can be used to manage the device. If no management address is explicitly configured to be advertised, the Brocade device will use the first available IPv4 address configured on the following types of interfaces, in the following order of preference:

- Physical port on which LLDP will be transmitting the packet
- Loopback interface
- Virtual routing interface (VE)
- Router interface on a VLAN that the port is a member of
- Other physical interface

If no IP address is configured, the port's current MAC address will be advertised.

The management address will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Management address (IPv4): 209.157.2.1
```

Port description

The port description TLV identifies the port from which the LLDP agent transmitted the advertisement. The port description is taken from the ifDescr MIB object from MIB-II.

By default, the port description is automatically advertised when LLDP is enabled on a global basis. To disable advertisement of the port description, enter a command such as the following.

```
FastIron(config)#no lldp advertise port-description ports e 2/4 to 2/12
```

The port description will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Port description: "GigabitEthernet20"
```

Syntax: [no] lldp advertise port-description ports ethernet <*slotnum/portnum*> | all

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System capabilities

The system capabilities TLV identifies the primary functions of the device and indicates whether these primary functions are enabled. The primary functions can be one or more of the following (more than one for example, if the device is both a bridge and a router):

- Repeater
- Bridge
- WLAN access point
- Router
- Telephone
- DOCSIS cable device
- Station only (devices that implement end station capability)
- Other

System capabilities for Brocade devices are based on the type of software image in use.

By default, the system capabilities are automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
FastIron(config)#no lldp advertise system-capabilities ports e 2/4 to 2/12
```

The system capabilities will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
System capabilities :   bridge
Enabled capabilities:   bridge
```

Syntax: [no] lldp advertise system-capabilities ports ethernet <*slotnum/portnum*> | all

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System description

The system description is the network entity, which can include information such as the product name or model number, the version of the system's hardware type, the software operating system level, and the networking software version. The information corresponds to the sysDescr MIB object in MIB-II.

To advertise the system description, enter a command such as the following.

```
FastIron(config)#lldp advertise system-description ports e 2/4 to 2/12
```

The system description will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
BigIron RX          #show lldp local-info
Local port: 1/2
  + Chassis ID (MAC address): 000c.dbf5.c000
  + Port ID (MAC address): 000c.dbf5.c000
  + Time to live: 120 seconds
  + System name         : "rx4"
  + Port description    : "10GigabitEthernet1/2"
  + System capabilities : bridge, router
    Enabled capabilities: bridge, router
  + 802.3 MAC/PHY          : auto-negotiation supported, but disabled
    Operational MAU type   : 10GigBaseLR
  + Link aggregation: not capable
  + Maximum frame size: 9212 octets
  + Port VLAN ID: none
  + Management address (IPv4): 200.200.200.11
  + Management address (IPv4): 200.200.200.10
```

Syntax: [no] lldp advertise system-description ports ethernet <*slotnum/portnum*> | all

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System name

The system name is the system's administratively assigned name, taken from the sysName MIB object in MIB-II. The sysName MIB object corresponds to the name defined with the CLI command **hostname**.

By default, the system name is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
FastIron(config)#no lldp advertise system-name ports e 2/4 to 2/12
```

The system name will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
System name:  "BigIron RX"
```

Syntax: [no] lldp advertise system-name ports ethernet <*slotnum/portnum*> | all

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

802.1 capabilities

Except for the VLAN name, the Brocade device will advertise the following 802.1 attributes when LLDP is enabled on a global basis:

- VLAN name (not automatically advertised)
- Untagged VLAN ID

VLAN name

The VLAN name TLV contains the name and VLAN ID of a VLAN configured on a port. An LLDPDU may include multiple instances of this TLV, each for a different VLAN.

To advertise the VLAN name, enter a command such as the following.

```
FastIron(config)#lldp advertise vlan-name vlan 99 ports e 2/4 to 2/12
```

The VLAN name will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
VLAN name (VLAN 99): "Voice-VLAN-99"
```

Syntax: [no] lldp advertise vlan-name vlan <*vlan ID*> ports ethernet <*slotnum/portnum*> | all

For <*vlan ID*>, enter the VLAN ID to advertise.

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Untagged VLAN ID

The port VLAN ID TLV advertises the Port VLAN Identifier (PVID) that will be associated with untagged or priority-tagged frames. If the port is not an untagged member of any VLAN (i.e., the port is strictly a tagged port), the value zero will indicate that.

By default, the port VLAN ID is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
FastIron(config)#no lldp advertise port-vlan-id ports e 2/4 to 2/12
```

The untagged VLAN ID will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Port VLAN ID: 99
```

Syntax: [no] lldp advertise port-vlan-id ports ethernet <*slotnum/portnum*> | all

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

802.3 capabilities

Except for Power-through-MDI information, the Brocade device will advertise the following 802.3 attributes when LLDP is enabled on a global basis:

- Link aggregation information
- MAC/PHY configuration and status
- Maximum frame size

Link aggregation

The **link-aggregation** TLV indicates the following:

- Whether the link is capable of being aggregated
- Whether the link is currently aggregated
- The primary trunk port

Brocade devices advertise link aggregation information about standard link aggregation (LACP) as well as static trunk configuration.

By default, link-aggregation information is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
FastIron(config)#no lldp advertise link-aggregation ports e 2/12
```

Syntax: [no] lldp advertise link-aggregation ports ethernet <*slotnum/portnum*> | all

The link aggregation advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Link aggregation: not capable
```

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

MAC/PHY configuration status

The MAC/PHY configuration and status TLV includes the following information:

- Auto-negotiation capability and status
- Speed and duplex mode
- Flow control capabilities for auto-negotiation
- Port speed down-shift and maximum port speed advertisement
- If applicable, indicates if the above settings are the result of auto-negotiation during link initiation or of a manual set override action

The advertisement reflects the effects of the following CLI commands:

- speed-duplex
- flow-control
- gig-default
- link-config

By default, the MAC/PHY configuration and status information are automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
FastIron(config)#no lldp advertise mac-phy-config-status ports e 2/4 to 2/12
```

The MAC/PHY configuration advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
     + 802.3 MAC/PHY       : auto-negotiation enabled
       Advertised capabilities: 10baseT-HD, 10baseT-FD, 100baseTX-HD,
   100baseTX-FD,
       fdxSPause, fdxBPause, 1000baseT-HD, 1000baseT-FD
       Operational MAU type: 100BaseTX-FD
```

Syntax: [no] lldp advertise mac-phy-config-status ports ethernet <*slotnum/portnum*> | all

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Maximum frame size

The maximum frame size TLV provides the maximum 802.3 frame size capability of the port. This value is expressed in octets and includes the four-octet Frame Check Sequence (FCS). The default maximum frame size is 1522. The advertised value may change depending on whether the **aggregated-vlan** or **jumbo** CLI commands are in effect.

By default, the maximum frame size is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
FastIron(config)#no lldp advertise max-frame-size ports e 2/4 to 2/12
```

The maximum frame size advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
   Maximum frame size: 1522 octets
```

Syntax: [no] lldp advertise max-frame-size ports ethernet <*slotnum/portnum*> | all

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword all may cause undesirable effects on some ports. For example, if you configure all ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Displaying LLDP statistics and configuration settings

You can use the following CLI **show** commands to display information about LLDP settings and statistics:

- **show lldp** – Displays a summary of the LLDP configuration settings.
- **show lldp statistics** – Displays LLDP global and per-port statistics.
- **show lldp neighbors** – Displays a list of the current LLDP neighbors.
- **show lldp neighbors detail** – Displays the details of the latest advertisements received from LLDP neighbors.
- **show lldp local-info** – Displays the details of the LLDP advertisements that will be transmitted on each port.

This above **show** commands are described in this section.

LLDP configuration summary

To display a summary of the LLDP configuration settings on the device, enter the **show lldp** command at any level of the CLI.

The following shows an example report.

```
FastIron#show lldp
LLDP transmit interval           : 10 seconds
LLDP transmit hold multiplier    : 4  (transmit TTL: 40 seconds)
LLDP transmit delay              : 1 seconds
LLDP SNMP notification interval  : 5 seconds
LLDP reinitialize delay          : 1 seconds
LLDP maximum neighbors           : 392
LLDP maximum neighbors per port  : 4
```

Syntax: show lldp

The following table describes the information displayed by the **show lldp statistics** command.

This field...	Displays...
LLDP transmit interval	The number of seconds between regular LLDP packet transmissions.
LLDP transmit hold multiplier	The multiplier used to compute the actual time-to-live (TTL) value of an LLDP advertisement. The TTL value is the transmit interval multiplied by the transmit hold multiplier.
LLDP transmit delay	The number of seconds the LLDP agent will wait after transmitting an LLDP frame and before transmitting another LLDP frame.

This field...	Displays...
LLDP reinitialize delay	The minimum number of seconds the device will wait from when LLDP is disabled on a port, until a request to re-enable LLDP on that port will be honored.
LLDP maximum neighbors	The maximum number of LLDP neighbors for which LLDP data will be retained, per device.
LLDP maximum neighbors per port	The maximum number of LLDP neighbors for which LLDP data will be retained, per port.

LLDP statistics

The **show lldp statistics** command displays an overview of LLDP neighbor detection on the device, as well as packet counters and protocol statistics. The statistics are displayed on a global basis.

The following shows an example report.

```
FastIron#show lldp statistics
Last neighbor change time: 23 hours 50 minutes 40 seconds ago

Neighbor entries added          : 14
Neighbor entries deleted        : 5
Neighbor entries aged out       : 4
Neighbor advertisements dropped : 0

Port       Tx Pkts   Rx Pkts   Rx Pkts   Rx Pkts   Rx TLVs   Rx TLVs Neighbors
             Total     Total  w/Errors Discarded Unrecognz Discarded  Aged Out
1            60963     75179         0         0         0         0         4
2                0         0         0         0         0         0         0
3            60963     60963         0         0         0         0         0
4            60963    121925         0         0         0         0         0
5                0         0         0         0         0         0         0
6                0         0         0         0         0         0         0
7                0         0         0         0         0         0         0
8                0         0         0         0         0         0         0
9                0         0         0         0         0         0         0
10           60974         0         0         0         0         0         0
11               0         0         0         0         0         0         0
12               0         0         0         0         0         0         0
13               0         0         0         0         0         0         0
14               0         0         0         0         0         0         0
```

Syntax: show lldp statistics

NOTE
You can reset LLDP statistics using the CLI command **clear LLDP statistics**. Refer to "Resetting LLDP statistics" on page 272.

The following table describes the information displayed by the **show lldp statistics** command.

This field...	Displays...
Last neighbor change time	The elapsed time (in hours, minutes, and seconds) since a neighbor last advertised information. For example, the elapsed time since a neighbor was last added, deleted, or its advertised information changed.
Neighbor entries added	The number of new LLDP neighbors detected since the last reboot or since the last time the **clear lldp statistics** all command was issued.

This field...	Displays...
Neighbor entries deleted	The number of LLDP neighbors deleted since the last reboot or since the last time the **clear lldp statistics all** command was issued.
Neighbor entries aged out	The number of LLDP neighbors dropped on all ports after the time-to-live expired. Note that LLDP entries age out naturally when a port's cable or module is disconnected or when a port becomes disabled. However, if a disabled port is re-enabled, the system will delete the old LLDP entries.
Neighbor advertisements dropped	The number of valid LLDP neighbors the device detected, but could not add. This can occur, for example, when a new neighbor is detected and the device is already supporting the maximum number of neighbors possible. This can also occur when an LLDPDU is missing a mandatory TLV or is not formatted correctly.
Port	The local port number.
Tx Pkts Total	The number of LLDP packets the port transmitted.
Rx Pkts Total	The number of LLDP packets the port received.
Rx Pkts w/Errors	The number of LLDP packets the port received that have one or more detectable errors.
Rx Pkts Discarded	The number of LLDP packets the port received then discarded.
Rx TLVs Unrecognz	The number of TLVs the port received that were not recognized by the LLDP local agent. Unrecognized TLVs are retained by the system and can be viewed in the output of the **show LLDP neighbors** detail command or retrieved through SNMP.
Rx TLVs Discarded	The number of TLVs the port received then discarded.
Neighbors Aged Out	The number of times a neighbor's information was deleted because its TTL timer expired.

LLDP neighbors

The **show lldp neighbors** command displays a list of the current LLDP neighbors per port.

The following shows an example report.

```
FastIron#show lldp neighbors
Lcl Port Chassis ID      Port ID         Port Description        System Name
1        0004.1234.0fc0  0004.1234.0fc0  GigabitEthernet9/1      BigIron RX 32~
1        00e0.5201.4000  00e0.5201.4000  GigabitEthernet0/1/1    BigIron RX 4~
3        00e0.5211.0200  00e0.5211.0203  GigabitEthernet4        BigIron RX 4~
4        00e0.5211.0200  00e0.5211.0202  GigabitEthernet3        BigIron RX 16~
4        00e0.5211.0200  00e0.5211.0210  GigabitEthernet17       BigIron RX 4~
15       00e0.5211.0200  00e0.5211.020f  GigabitEthernet16       BigIron RX 8~
16       00e0.5211.0200  00e0.5211.020e  GigabitEthernet15       BigIron RX 16~
17       00e0.5211.0200  00e0.5211.0211  GigabitEthernet18       BigIron RX 4~
18       00e0.5211.0200  00e0.5211.0210  GigabitEthernet17       BigIron RX 4~
```

Syntax: show lldp neighbors

The following table describes the information displayed by the **show lldp neighbors** command.

This field...	Displays...
Lcl Port	The local LLDP port number.
Chassis ID	The identifier for the chassis. Brocade devices use the base MAC address of the device as the Chassis ID.

This field...	Displays...
Port ID	The identifier for the port. Brocade devices use the permanent MAC address associated with the port as the port ID.
Port Description	The description for the port. Brocade devices use the ifDescr MIB object from MIB-II as the port description.
System Name	The administratively-assigned name for the system. Brocade devices use the sysName MIB object from MIB-II, which corresponds to the CLI **hostname** command setting. **NOTE:** A tilde (~) at the end of a line indicates that the value in the field is too long to display in full and is truncated.

LLDP neighbors detail

The **show lldp neighbors detail** command displays the LLDP advertisements received from LLDP neighbors.

The following shows an example **show lldp neighbors detail** report.

NOTE
The **show lldp neighbors detail** output will vary depending on the data received. Also, values that are not recognized or do not have a recognizable format, may be displayed in hexadecimal binary form.

```
FastIron#show lldp neighbors detail ports e 1/9
Local port: 1/9
  Neighbor: 0800.0f18.cc03, TTL 101 seconds
    + Chassis ID (network address): 10.43.39.151
    + Port ID (MAC address): 0800.0f18.cc03
    + Time to live: 120 seconds
    + Port description    : "LAN port"
    + System name         : "regDN 1015,MITEL 5235 DM"
    + System description  : "regDN 1015,MITEL 5235 DM,h/w rev 2,ASIC rev 1,f/w\
                             Boot 02.01.00.11,f/w Main 02.01.00.11"
    + System capabilities : bridge, telephone
      Enabled capabilities: bridge, telephone
    + Management address (IPv4): 10.43.39.151
    + 802.3 MAC/PHY          : auto-negotiation enabled
      Advertised capabilities: 10BaseT-HD, 10BaseT-FD, 100BaseTX-HD,
                               100BaseTX-FD
      Operational MAU type   : 100BaseTX-FD
    + MED capabilities: capabilities, networkPolicy, extendedPD
      MED device type : Endpoint Class III
    + MED Network Policy
      Application Type  : Voice
      Policy Flags      : Known Policy, Tagged
      VLAN ID           : 300
      L2 Priority       : 7
      DSCP Value        : 7
    + MED Extended Power through MDI
      Power Type     : PD device
      Power Source   : Unknown Power Source
      Power Priority : High (2)
      Power Value    : 6.2 watts (PSE equivalent: 6656 mWatts)
    + MED Hardware revision : "PCB Version: 2"
    + MED Firmware revision : "Boot 02.01.00.11"
    + MED Software revision : "Main 02.01.00.11"
    + MED Serial number     : ""
    + MED Manufacturer      : "Mitel Corporation"
    + MED Model name        : "MITEL 5235 DM"
    + MED Asset ID          : ""
```

A backslash (\) at the end of a line indicates that the text continues on the next line.

Except for the following field, the fields in the above output are described in the individual TLV advertisement sections in this chapter.

This field...	Displays...
Neighbor	The source MAC address from which the packet was received, and the remaining TTL for the neighbor entry.

Syntax: show lldp neighbors detail [ports ethernet <*slotnum/portnum*> | all]

If you do not specify any ports or use the keyword **all**, by default, the report will show the LLDP neighbor details for all ports.

You can list all of the ports individually, use the keyword to to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword all instead of listing the ports individually.

LLDP configuration details

The **show lldp local-info** command displays the local information advertisements (TLVs) that will be transmitted by the LLDP agent.

NOTE
The **show lldp local-info** output will vary based on LLDP configuration settings.

The following shows an example report.

```
BigIron RX#show lldp local-info ports e 20/1
Local port: 20/1
  + Chassis ID (MAC address): 0012.f233.e2c0
  + Port ID (MAC address): 0012.f233.e2d3
  + Time to live: 40 seconds
  + System name: "FESX424_POE"
  + Port description: "GigabitEthernet20"
  + System description  : "Foundry Networks, Inc. FESX424-PREM-PoE, IronWare V\
                           ersion 04.0.00b256T3e1 Compiled on Sep 04 2007 at 0\
                           3:54:29 labeled as SXS04000b256"
  + System capabilities : bridge
    Enabled capabilities: bridge
  + 802.3 MAC/PHY       : auto-negotiation enabled
    Advertised capabilities: 10BaseT-HD, 10BaseT-FD, 100BaseTX-HD,
                             100BaseTX-FD, fdxSPause, fdxBPause, 1000BaseT-HD,
                             1000BaseT-FD
    Operational MAU type: 100BaseTX-FD
  + 802.3 Power through MDI: PSE port, power enabled, class 2
    Power Pair         : A (not controllable)
  + Link aggregation: not capable
  + Maximum frame size: 1522 octets
  + MED capabilities: capabilities, networkPolicy, location, extendedPSE
    MED device type : Network Connectivity
  + MED Network Policy
    Application Type  : Voice
    Policy Flags      : Known Policy, Tagged
    VLAN ID           : 99
    L2 Priority       : 3
    DSCP Value        : 22
  + Management address (IPv4): 192.1.1.121
  + VLAN name (VLAN 99): "Voice-VLAN-99"
```

A backslash (\) at the end of a line indicates that the text continues on the next line.

The fields in the above output are described in the individual TLV advertisement sections in this chapter.

Syntax: show lldp local-info [ports ethernet <*slot num/port num*> | all]

If you do not specify any ports or use the keyword **all**, by default, the report will show the local information advertisements for all ports.

You can list all of the ports individually, use the keyword to to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword all instead of listing the ports individually.

Resetting LLDP statistics

To reset LLDP statistics, enter the **clear lldp statistics** command at the Global CONFIG level of the CLI. The Brocade device will clear the global and per-port LLDP neighbor statistics on the device (refer to "LLDP statistics" on page 267).

```
FastIron#clear lldp statistics
```

Syntax: clear lldp statistics [ports ethernet <*slot num/port num*> | all]

If you do not specify any ports or use the keyword **all**, by default, the system will clear lldp statistics on all ports.

You can list all of the ports individually, use the keyword to to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword all instead of listing the ports individually.

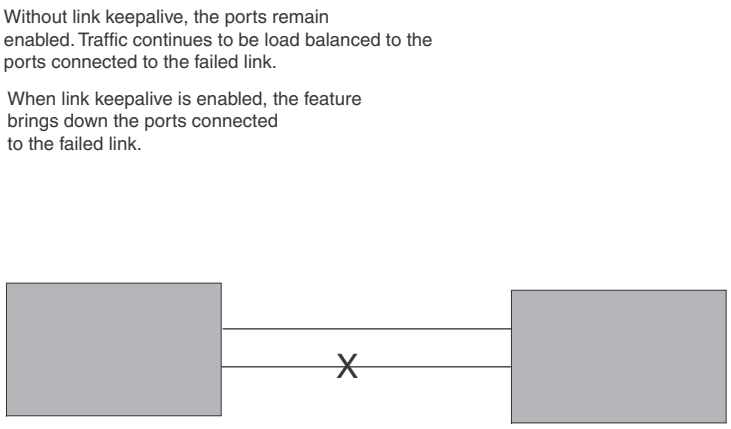
Chapter

Configuring Uni-Directional Link Detection (UDLD) 10

In this chapter

This chapter describes configuring Uni-Directional Link Detection.Uni-directional Link Detection (UDLD) monitors a link between two device 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 trunk links. Figure 20 shows an example.

FIGURE 20 UDLD example



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 two more intervals. If the port still does not receive a health-check packet after waiting for three intervals, UDLD will be kept in a suspended state until it receives the first keep-alive message from the other end. In this suspended state, UDLD will continue to send the keep-alive message but will not bring the port down after maximum number of retries is done and no keep-alive message is received from the other end. The UDLD will transition from this suspended state to active state after it receives the first keep-alive message from the other end. In the active state, UDLD peers will continue to exchange keep alive messages periodically and if there are keep-alive messages are missed for certain number of times from the other end, UDLD will bring down the logical port. The UDLD will then transition from active to suspended state.

Everytime UDLD is enabled on a port, the port will be transitioned into the suspended state to detect if the other end (peer) supports UDLD. This include the case where:

- User enables UDLD on a port
- A port that has UDLD enabled coming back up after an system reboot
- Port was brought down by UDLD after uni-directional link was detected and now the problem is fixed.

Configuration considerations

- The feature is supported only on Ethernet ports.
- To configure UDLD on a trunk group, you must configure the feature on each port of the group individually. Configuring UDLD on a trunk group's primary port enables the feature on that port only.
- Dynamic trunking is not supported. If you want to configure a trunk group that contains ports on which UDLD is enabled, you must remove the UDLD configuration from the ports. After you create the trunk 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.

```
BigIron RX(config)# link-keepalive ethernet 1/1
```

Syntax: [no] link-keepalive ethernet <*slot*>/<*portnum*> [ethernet <*slot*>/<*portnum*>]

To enable the feature on a trunk group, enter commands such as the following.

```
BigIron RX(config)# link-keepalive ethernet 1/1 ethernet 1/2
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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.

When UDLD is enabled on a port, The UDLD starts sending the keep-alive messages at a preconfigured interval. In the current implementation, if there is no keep-alive received from the other end of this link after 3 retries then this port is set to logical link down. With the new design, after the UDLD is enabled on a port, UDLD will be kept in a newly created suspended state until it receives first keep-alive message from the other end. In this suspended state, UDLD will continue to send the keep-alive message but will not bring the port down after maximum number of retries is done and no keep-alive message is received from the other end. The UDLD will transition from this suspended state to active state after it receives the first keep-alive message from the other end. In the active state, UDLD peers will continue to exchange keep alive messages periodically and if there are keep-alive messages are missed for certain number of times from the other end, UDLD will bring down the logical port. The UDLD will then transition from active to suspended state.

Displaying UDLD information

Displaying information for all ports

To display UDLD information for all ports, enter the following command.

```
BigIron RX(config)# show link-keepalive
Total link-keepalive enabled ports: 4
Keepalive Retries: 5    Keepalive Interval: 1 Sec.

Port    Physical Link   Link-keepalive    Logical Link
4/1     up              up                up
4/2     up              up                up
4/3     down            down              down
4/4     up              down              down
```

Syntax: show link-keepalive [ethernet <*slot*>/<*portnum*>]

Displaying link-keepalive information

The **show link-keepalive** command will indicate the physical link and logical link as "UP" and the link-keepalive as "init" when the first enabled link-keepalive on one end, and while its connected peer has not enabled UDLD yet.

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.

```
BigIron RX(config)#sh link-keepalive
Total link-keepalive enabled ports: 2
Keepalive Retries: 5    Keepalive Interval: 5 * 100 MilliSec.

Port    Physical Link   Link-keepalive  Logical link
1/15    up              init            up
2/15    up              init            up
```

Syntax: show link-keepalive

TABLE 60 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 device port and the directly connected device.
Link-keepalive	Show if the keepalive link is up or down.
Logical Link	The state of the logical link. This is the state of the link between this device port and the device 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".

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.

```
BigIron RX(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 interfaces brief" is an abbreviation of "show ip interfaces brief". The "ip" is implied. If a ve interface has no IP address and only an address in a different protocol, "show interfaces brief" will show the interface as down. This is by design. To view ve interface status in relation to protocols other than IP, please specifify the protocol. For example, "show ipx interface ve <*num*>" or "show appletalk interface ve <*num*>".

The **show link-keepalive** command shows the following.

```
BigIron RX(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, enter a command such as the following.

```
BigIron RX(config)# show link-keepalive ethernet 4/1

Current State   : up            Remote MAC Addr  : 00e0.52d2.5100
Local Port      : 4/1           Remote Port      : 2/1
Local System ID : e0927400      Remote System ID : e0d25100
Packets sent    : 254           Packets received : 255
Transitions     : 1
```

TABLE 61 CLI display of detailed UDLD information

This field...	Displays...
Current State	The state of the logical link. This is the link between this device port and the device 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 device.
Remote Port	The port number on the device at the remote end of the link.
Local System ID	A unique value that identifies this device. The ID can be used by Brocade technical support for troubleshooting.
Remote System ID	A unique value that identifies the device 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.
Port blocking	Information used by Brocade technical support for troubleshooting.

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.

```
BigIron RX(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.

```
BigIron RX# 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

VLANs 11

In this chapter

Overview of Virtual Local Area Networks (VLANs)

Virtual Local Area Networks (VLANs) allow you to segment traffic in a network by placing ports and interfaces into separate broadcast domains. Each broadcast domain is uniquely identified by VLAN IDs. These broadcast domains can span multiple devices.

The device supports two types of VLANs: *port-based VLANs* and *protocol-based VLANs*. A port-based VLAN consists of interfaces that constitutes a Layer 2 broadcast domain. By default, all interfaces on a BigIron RX 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 port assigned is untagged. If the port assigned is tagged, then the port remains as untag on the original vlan (vlan1) and behaves as dual-mode port.

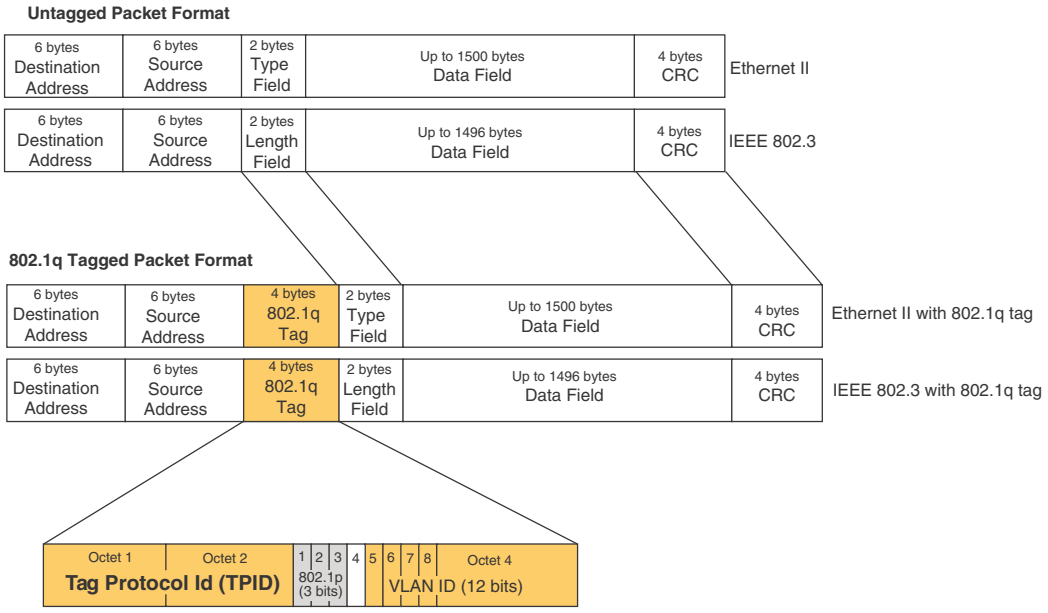
Tagged, untagged, and dual-mode ports

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 device 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 802.1q specification. You can change this tag value on a global basis on device if needed to be compatible with other vendors' equipment.

Figure 21 shows the format of packets with and without the 802.1q tag. The tag format is vendor-specific. To use the tag for VLANs configured across multiple devices, make sure all the devices support the same tag format.

FIGURE 21 Packet containing *Brocade*'s 802.1QVLAN tag

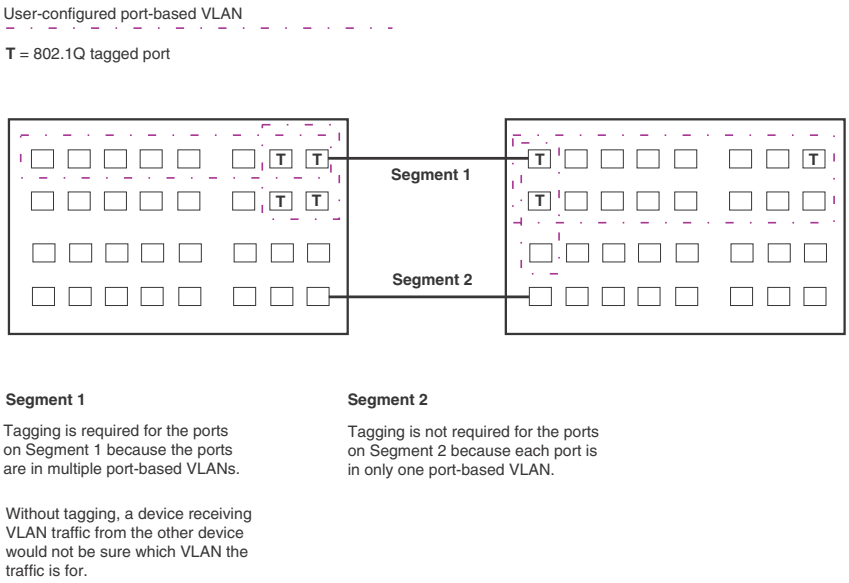


If you configure a VLAN that spans multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If a port connecting one device to the other is a member of only a single port-based VLAN, tagging is not required.

If you use tagging on multiple devices, each device must be configured for tagging and must use the same tag value. In addition, the implementation of tagging must be compatible on the devices. The tagging on all device switches is compatible with other *Brocade* devices.

Figure 22 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 22 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 using protocol-based VLANs. Protocol-based VLANs accept broadcasts of a specified protocol type. For example, an IP subnet VLAN accepts only broadcasts for the specified IP subnets. This feature enables you to limit the amount of broadcast traffic to end-stations, servers, and routers.

In a device, 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 device receives an IPv6 multicast packet (a packet with 06 in the version field and 0xFF as the beginning of the destination address), the device 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 support on the device.

If there are ports in a port-based VLAN that you want to exclude from protocol-based VLANs, the protocol-based VLAN can be configured to explicitly exclude those ports.

VLAN configuration rules

To create any type of VLAN on a device, Layer 2 forwarding must be enabled. When Layer 2 forwarding is enabled, the device becomes a switch on all ports for all non-routable protocols.

The device can only support up to 254 independent VLAN with Layer 2 protocols.

In addition to this rule, the sections below summarize the rules for configuring VLANs.

VLAN ID range

VLAN IDs can be one of the following: 1 – 4089. IDs 4090 – 4094 are reserved for 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 device 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

Given below are the membership rules for multiple VLAN:

- 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 device.
- Removing a configured port-based VLAN from a device automatically removes any protocol-based VLAN, or any virtual routing interfaces defined within the port-based VLAN.

Layer 2 control protocols on VLANs

Layer 2 protocols such as STP, RSTP, MRP, and VSRP can be enabled on a port-based VLANs, but you cannot enable or disable these protocols for protocol-based VLANs.

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.

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.

To create a VLAN, do the following.

1. At the global CONFIG level assign an ID to the VLAN. For example,

   ```
   BigIron RX(config)# vlan 2
   ```

   Syntax: [no] vlan-id [name <*vlan-name*>]

   VLAN IDs can be in the range of 1 – 4089; however, do not use VLANs 4090 – 4094. These IDs are reserved and are used for control purposes. Also, VLAN IDs 0 and 4095 are reserved by the IEEE standards and cannot be configured. Use the **no** form of the command to delete the VLAN from the configuration.

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 an 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.

```
BigIron RX(config-vlan-2)# untag e 1/9 to 1/16
BigIron RX(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 explicitly 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 device 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 trunk group ports to the VLAN by entering the trunk group's the primary port. A trunk group's primary port is the port with the lowest number in the trunk group. When you add the trunk group's primary port, all the ports on the trunk group become members of the VLAN.

Use the **no** form of the command to remove the ports from a VLAN. For example.

```
BigIron RX(config)# vlan 4
BigIron RX(config-vlan-4)# no untag ethernet 1/11
```

VLAN byte accounting

To enable your device to perform accounting of the number of bytes received by all the member ports of a VLAN. This includes the preamble and the minimum inter-frame gap in Ethernet. The byte counts can then be viewed using the **show vlan** command. VLAN byte accounting is disabled by default.

Considerations when configuring VLAN byte accounting

- VLAN byte accounting cannot be enabled for the default or control VLANs.
- The number of VLANs on which byte accounting can be enabled system-wide is restricted by the number of VLANs with byte accounting enabled on a given packet processor and the number of rate limiting policies enabled on the same packet processor ports. Refer to Table 62 for details.

- On a given packet processor, the total number of VLANs with byte accounting enabled and the number of ACL-based and VLAN-based rate limiting policies is dependent on the interface module. Refer to Table 62 for details.
- If a port's VLAN has byte accounting enabled, you cannot enable rate limiting on that port. Similarly, if a port has rate limiting enabled, you cannot enable VLAN byte accounting on that port's VLAN.
- Clearing the rate limiting counters using **clear rate-limit counters** will also clear VLAN byte-accounting counters. It is recommended that when using rate limiting along with VLAN byte accounting, use individual port rate limiting counters.

Configuring VLAN byte accounting

To enable VLAN accounting on a specified VLAN, use the following commands.

```
BigIron RX(config)# vlan 10
BigIron RX(config-vlan-10)# byte-accounting
```

Syntax: [no] byte-accounting

Displaying VLAN byte accounting information

To display VLAN accounting information for all VLANs configured on a router, use the **show vlan** command as shown.

```
BigIron RX# show vlan
Configured PORT-VLAN entries: 2
Maximum PORT-VLAN entries: 512
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
L2 protocols : NONE
Untagged Ports : ethe 1/1 to 1/40 ethe 2/1 to 2/4
PORT-VLAN 10, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/2 to 1/5
Bytes received : 18527
```

To display VLAN accounting information for a specific VLAN, use the **show vlan <vlan>** command as shown.

```
BigIron RX# show vlan 10
PORT-VLAN 10, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/2 to 1/5
Bytes received : 5626
```

The Bytes received field displays the number of bytes received by all member ports of all VLANs configured on the router.

Maximum number of rate limiting policies and VLANs with byte accounting

The maximum number of ACL-based, and VLAN-based rate limiting policies that can be configured on ports controlled by the same packet processor also depends on the number of VLANs with byte accounting enabled on the same packet processor.

On a given packet processor (PPCR), the total of:

Number of VLANs with byte accounting enabled
+
Number of rate limiting policies based on ACLs and VLANs
cannot exceed the maximum number of policies as specified in Table 62.

TABLE 62 Maximum # of rate limiting policies and VLANs w/ byte accounting permitted per-PPCR

Module type	PPCR number	Port #	Max # of rate limiting policies based on ACLs and VLANs + number of VLANs w/ byte accounting enabled
24 x 1G	PPCR 1	1 - 12	115
	PPCR 2	13 - 24	115

Clearing counters

To clear the byte counter for a VLAN, enter a command such as the following.

```
BigIron RX(config) #clear vlan byte-accounting 10
```

You can also enter the following command to clear the byte counter for all VLANs.

```
BigIron RX(config) #clear vlan byte-accounting all-vlans
```

Syntax: clear vlan byte-accounting <*vlan-id*> | all-vlans

Enter a VLAN ID if you want to clear the byte counters for a specific VLAN. Enter **all-vlans** to clear the byte counters for all VLANs.

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.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# tagged e 1/1 to 1/8
BigIron RX(config-vlan-2)# vlan 1
BigIron RX(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.

```
BigIron RX(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 device 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.

```
BigIron RX(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. Valid VLAN IDs are from 1 – 4089; however, do not use VLANs 4090 – 4094, which are reserved for 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.

   ```
   BigIron RX(config)# vlan 2
   BigIron RX(config-vlan-2)# untag e 1/9 to 1/16
   BigIron RX(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.

   ```
   BigIron RX(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

   ```
   BigIron RX(config-vlan-group-ipv6-proto)# static e 1/1 e 1/24
   BigIron RX(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 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.

```
BigIron RX(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 **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 virtual routing interfaces

The device 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 device 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 device 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.

For example,

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# tagged e 1/1 to 1/2
BigIron RX(config-vlan-2)# ip-proto
BigIron RX(config-vlan-group-ip-proto)# router-interface ve 1
```

The device can locally route IP packets between VLANs that are defined within a single router. All other routable protocols or protocol-based VLANs (for example, IPX and AppleTalk) must be routed by another external router capable of routing the protocol.

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.

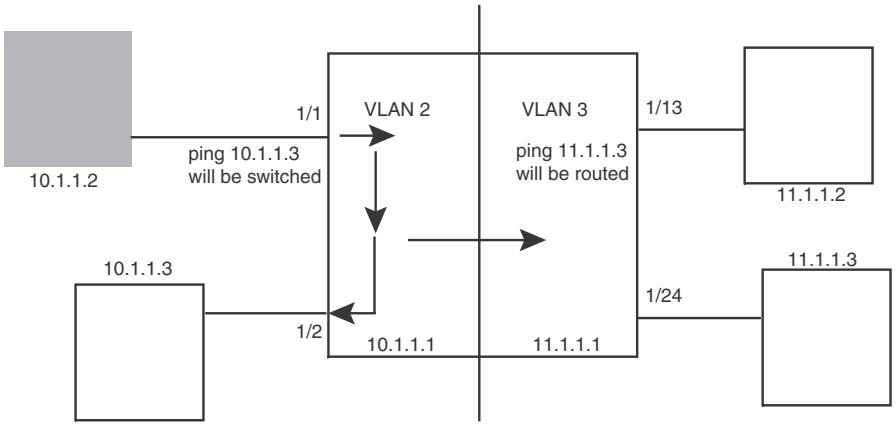
```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# tagged e 1/1 to 1/2
BigIron RX(config-vlan-2)# router-interface ve 2
BigIron RX(config-vlan-2)# exit
BigIron RX(config)# interface ve 2
BigIron RX(config-ve-2)# ip address 10.1.1.1/24
```

Syntax: router-interface ve <*ve-number*>

Enter 1 to the maximum number of virtual routing interfaces supported on the device for <*ve-number*>.

Bridging and routing the same protocol simultaneously on the same device

Some configurations may require simultaneous switching and routing of the same single protocol across different sets of ports on the same router. When IP routing is enabled on a device, you can route IP packets on specific interfaces while bridging them on other interfaces. In this scenario, you can create two separate backbones for the same protocol, one bridged and one routed.



The following is a sample configuration for the illustration above.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# tagged e 1/1 to 1/2
BigIron RX(config-vlan-2)# router-inter ve 2
BigIron RX(config-vlan-2)# ip-proto static e 1/1/ to 1/2
BigIron RX(config-vlan-2)# exit
BigIron RX(config)# vlan 3
BigIron RX(config-vlan-3)# tagged e 1/13 to 1/24
BigIron RX(config-vlan-3)# router-int ve 3
BigIron RX(config-vlan-3)# exit
BigIron RX(config)# interface ve 2
BigIron RX(config-ve-2)# ip address 10.1.1.1/24
BigIron RX(config-if-e1000-2/1)# exit
BigIron RX(config)# interface ve 3
BigIron RX(config-ve-3)# ip address 11.1.1.2/24
```

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.

Integrated Switch Routing (ISR)

Brocade **Integrated Switch Routing (ISR)** feature enables VLANs configured on the device 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 device, 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 device forwards IP broadcasts within each VLAN at Layer 2 but routes Layer 3 traffic between the VLANs using the virtual routing interfaces.

NOTE
The device 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 device 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 is consisted 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 device 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 Brocade' very powerful Integrated Switch Routing (ISR) technology. ISR is very flexible and can solve many networking problems.

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.

You can create up to 32 VLAN groups

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 311.

Configuring a VLAN group

To configure a VLAN group, do the following.

1. Create the VLAN group and assign the VLANs to that group.

   ```
   BigIron RX(config)# vlan-group 1 vlan 2 to 1000
   ```

   Syntax: [no] vlan-group <*num*> vlan <*vlan-id*> to <*vlan-id*>

   Use 1 – 32 for <*num*> parameter with the **vlan-group** command specifies the VLAN group ID and can be from.

   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 311.

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.

```
BigIron RX(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*>]

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.

```
BigIron RX(config-vlan-group-1)# add-vlan 1001 to 1002
BigIron RX(config-vlan-group-1)# remove-vlan 900 to 1000
```

Syntax: [no] add-vlan <*vlan-id*> [to <*vlan-id*>]

Syntax: 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.

```
BigIron RX(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 ethe 1/1 to 1/2
```

Displaying information about VLAN groups

To display VLAN group configuration information, enter the following command.

```
BigIron RX# 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: ethe 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.

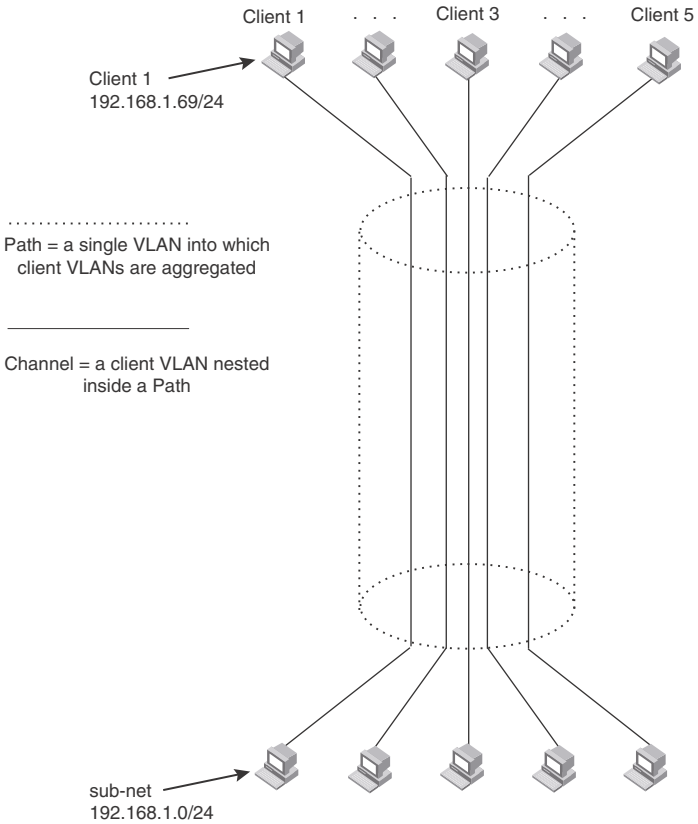
You can aggregate up to 4089 VLANs within another VLAN. This provides a total VLAN capacity on one device of 16,760,836 channels (4089 * 4089).

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) 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 23 shows a conceptual picture of the service that aggregated VLANs provide.

FIGURE 23 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 24 shows an example application that uses aggregated VLANs. This configuration includes the client connections shown in Figure 23.

FIGURE 24 Example super aggregated VLAN application

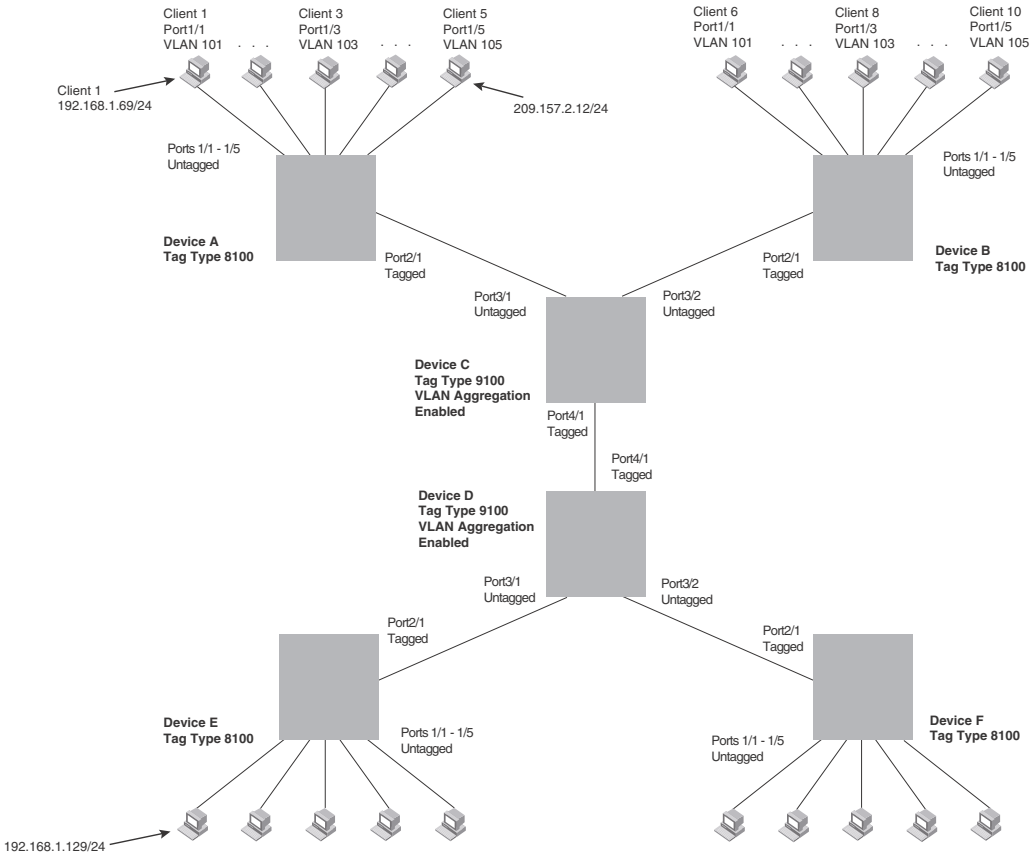


Figure 22 shows a collocation service provides private channels for multiple clients. Although the same devices are used for all the clients, the VLANs ensure that each client receives its own Layer 2 broadcast domain, separate from the broadcast domains of other clients. For example, client 1 cannot 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 trunk 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 following tasks.

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 24 on page 294, enter commands such as the following.

```
BigIron RX(config)# vlan 101
BigIron RX(config-vlan-101)# tagged ethernet 2/1
BigIron RX(config-vlan-101)# untagged ethernet 1/1
BigIron RX(config-vlan-101)# exit
BigIron RX(config)# vlan 102
BigIron RX(config-vlan-102)# tagged ethernet 2/1
BigIron RX(config-vlan-102)# untagged ethernet 1/2
BigIron RX(config-vlan-102)# exit
BigIron RX(config)# vlan 103
BigIron RX(config-vlan-103)# tagged ethernet 2/1
BigIron RX(config-vlan-103)# untagged ethernet 1/3
BigIron RX(config-vlan-103)# exit
BigIron RX(config)# vlan 104
BigIron RX(config-vlan-104)# tagged ethernet 2/1
BigIron RX(config-vlan-104)# untagged ethernet 1/4
BigIron RX(config-vlan-104)# exit
BigIron RX(config)# vlan 105
BigIron RX(config-vlan-105)# tagged ethernet 2/1
BigIron RX(config-vlan-105)# untagged ethernet 1/5
BigIron RX(config-vlan-105)# exit
BigIron RX(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:

- Enable VLAN aggregation. This support allows the core device to add an additional tag to each Ethernet frame that contains a VLAN packet from the edge device. The additional tag identifies the aggregate VLAN (the path). However, the additional tag can cause the frame to be longer than the maximum supported frame size. The larger frame support allows Ethernet frames up to 1530 bytes long.

NOTE
Enable the VLAN aggregation option only on the core devices.

- 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 24 on page 294, enter the following commands.

```
BigIron RX(config)# tag-type 9100
BigIron RX(config)# aggregated-vlan
BigIron RX(config)# vlan 101
BigIron RX(config-vlan-101)# tagged ethernet 4/1
BigIron RX(config-vlan-101)# untagged ethernet 3/1
BigIron RX(config-vlan-101)# exit
BigIron RX(config)# vlan 102
BigIron RX(config-vlan-102)# tagged ethernet 4/1
BigIron RX(config-vlan-102)# untagged ethernet 3/2
BigIron RX(config-vlan-102)# exit
BigIron RX(config)# write memory
```

Syntax: **[no]** tag-type <*num*>

Syntax: **[no]** aggregated-vlan

The <*num*> parameter specifies the tag type. It can be a hexadecimal value from 0 – ffff. The default is 8100.

Complete CLI examples

The following sections show all the Aggregated VLAN configuration commands on the devices in Figure 24 on page 294.

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 24 on page 294 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

```
BigIron RX-A(config)# vlan 101
BigIron RX-A(config-vlan-101)# tagged ethernet 2/1
BigIron RX-A(config-vlan-101)# untagged ethernet 1/1
BigIron RX-A(config-vlan-101)# exit
BigIron RX-A(config)# vlan 102
BigIron RX-A(config-vlan-102)# tagged ethernet 2/1
BigIron RX-A(config-vlan-102)# untagged ethernet 1/2
BigIron RX-A(config-vlan-102)# exit
BigIron RX-A(config)# vlan 103
BigIron RX-A(config-vlan-103)# tagged ethernet 2/1
BigIron RX-A(config-vlan-103)# untagged ethernet 1/3
BigIron RX-A(config-vlan-103)# exit
BigIron RX-A(config)# vlan 104
BigIron RX-A(config-vlan-104)# tagged ethernet 2/1
BigIron RX-A(config-vlan-104)# untagged ethernet 1/4
BigIron RX-A(config-vlan-104)# exit
BigIron RX-A(config)# vlan 105
BigIron RX-A(config-vlan-105)# tagged ethernet 2/1
BigIron RX-A(config-vlan-105)# untagged ethernet 1/5
BigIron RX-A(config-vlan-105)# exit
BigIron RX-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.

```
BigIron RX-B(config)# vlan 101
BigIron RX-B(config-vlan-101)# tagged ethernet 2/1
BigIron RX-B(config-vlan-101)# untagged ethernet 1/1
BigIron RX-B(config-vlan-101)# exit
BigIron RX-B(config)# vlan 102
BigIron RX-B(config-vlan-102)# tagged ethernet 2/1
BigIron RX-B(config-vlan-102)# untagged ethernet 1/2
BigIron RX-B(config-vlan-102)# exit
BigIron RX-B(config)# vlan 103
BigIron RX-B(config-vlan-103)# tagged ethernet 2/1
BigIron RX-B(config-vlan-103)# untagged ethernet 1/3
BigIron RX-B(config-vlan-103)# exit
BigIron RX-B(config)# vlan 104
BigIron RX-B(config-vlan-104)# tagged ethernet 2/1
BigIron RX-B(config-vlan-104)# untagged ethernet 1/4
BigIron RX-B(config-vlan-104)# exit
BigIron RX-B(config)# vlan 105
BigIron RX-B(config-vlan-105)# tagged ethernet 2/1
BigIron RX-B(config-vlan-105)# untagged ethernet 1/5
BigIron RX-B(config-vlan-105)# exit
BigIron RX-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.

```
BigIron RX-C(config)# tag-type 9100
BigIron RX-C(config)# aggregated-vlan
BigIron RX-C(config)# vlan 101
BigIron RX-C(config-vlan-101)# tagged ethernet 4/1
BigIron RX-C(config-vlan-101)# untagged ethernet 3/1
BigIron RX-C(config-vlan-101)# exit
BigIron RX-C(config)# vlan 102
BigIron RX-C(config-vlan-102)# tagged ethernet 4/1
BigIron RX-C(config-vlan-102)# untagged ethernet 3/2
BigIron RX-C(config-vlan-102)# exit
BigIron RX-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.

```
BigIron RX-D(config)# tag-type 9100
BigIron RX-D(config)# aggregated-vlan
BigIron RX-D(config)# vlan 101
BigIron RX-D(config-vlan-101)# tagged ethernet 4/1
BigIron RX-D(config-vlan-101)# untagged ethernet 3/1
BigIron RX-D(config-vlan-101)# exit
BigIron RX-D(config)# vlan 102
BigIron RX-D(config-vlan-102)# tagged ethernet 4/1
BigIron RX-D(config-vlan-102)# untagged ethernet 3/2
BigIron RX-D(config-vlan-102)# exit
BigIron RX-D(config)# write memory
```

Commands for device E

Since the configuration in Figure 24 on page 294 is symmetrical, the commands for configuring device E are identical to the commands for configuring device A.

```
BigIron RX-E(config)# vlan 101
BigIron RX-E(config-vlan-101)# tagged ethernet 2/1
BigIron RX-E(config-vlan-101)# untagged ethernet 1/1
BigIron RX-E(config-vlan-101)# exit
BigIron RX-E(config)# vlan 102
BigIron RX-E(config-vlan-102)# tagged ethernet 2/1
BigIron RX-E(config-vlan-102)# untagged ethernet 1/2
BigIron RX-E(config-vlan-102)# exit
BigIron RX-E(config)# vlan 103
BigIron RX-E(config-vlan-103)# tagged ethernet 2/1
BigIron RX-E(config-vlan-103)# untagged ethernet 1/3
BigIron RX-E(config-vlan-103)# exit
BigIron RX-E(config)# vlan 104
BigIron RX-E(config-vlan-104)# tagged ethernet 2/1
BigIron RX-E(config-vlan-104)# untagged ethernet 1/4
BigIron RX-E(config-vlan-104)# exit
BigIron RX-E(config)# vlan 105
```

```
BigIron RX-E(config-vlan-105)# tagged ethernet 2/1
BigIron RX-E(config-vlan-105)# untagged ethernet 1/5
BigIron RX-E(config-vlan-105)# exit
BigIron RX-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 24 on page 294 are symmetrical, the configuration of device F is also identical to the configuration of device A and device B.

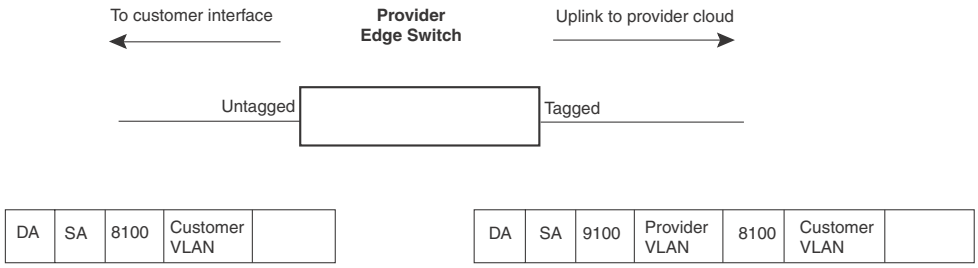
```
BigIron RX-F(config)# vlan 101
BigIron RX-F(config-vlan-101)# tagged ethernet 2/1
BigIron RX-F(config-vlan-101)# untagged ethernet 1/1
BigIron RX-F(config-vlan-101)# exit
BigIron RX-F(config)# vlan 102
BigIron RX-F(config-vlan-102)# tagged ethernet 2/1
BigIron RX-F(config-vlan-102)# untagged ethernet 1/2
BigIron RX-F(config-vlan-102)# exit
BigIron RX-F(config)# vlan 103
BigIron RX-F(config-vlan-103)# tagged ethernet 2/1
BigIron RX-F(config-vlan-103)# untagged ethernet 1/3
BigIron RX-F(config-vlan-103)# exit
BigIron RX-F(config)# vlan 104
BigIron RX-F(config-vlan-104)# tagged ethernet 2/1
BigIron RX-F(config-vlan-104)# untagged ethernet 1/4
BigIron RX-F(config-vlan-104)# exit
BigIron RX-F(config)# vlan 105
BigIron RX-F(config-vlan-105)# tagged ethernet 2/1
BigIron RX-F(config-vlan-105)# untagged ethernet 1/5
BigIron RX-F(config-vlan-105)# exit
BigIron RX-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 trunk ports, thereby enabling the creation of two identical 802.1Q tags (802.1Q-in-Q tagging) on a single device. This feature improves SAV interoperability between *Brocade* devices and other vendors' devices that support the 802.1Q tag-types, but are not very flexible with the tag-types they accept.

Figure 25 on page 299 shows an 802.1Q configuration example.

FIGURE 25 802.1Q configuration example

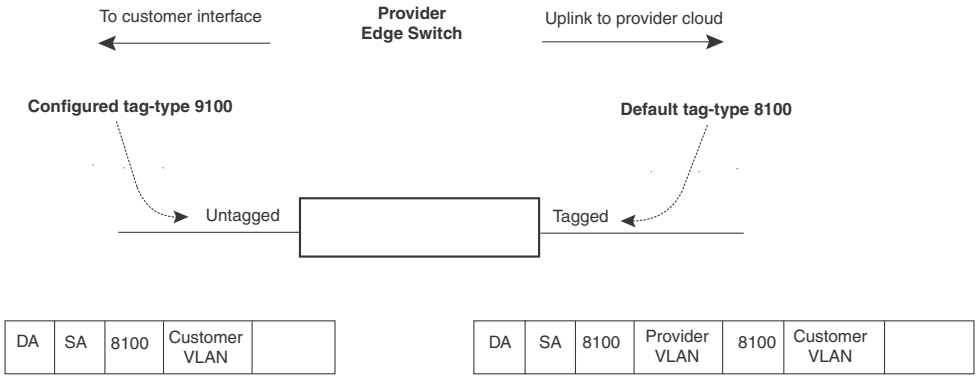


As shown in Figure 25, the ports to customer interfaces are untagged, whereas the uplink ports to the provider cloud are tagged, because multiple client VLANs share the uplink to the provider cloud. In this example, the device treats the customer's private VLAN ID and 8100 tag type as normal payload, 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 26 shows an example application of the 802.1Q-in-Q enhancement.

FIGURE 26 802.1Q-in-Q configuration example



In Figure 26, 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 device 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:

- Since the uplink (to the provider cloud) and the edge link (to the customer port) must have different 802.1Q tags, make sure the uplink and edge link are in different port regions.
- If you configure a port with an 802.1Q tag-type, the device automatically applies the 802.1Q tag-type to all ports within the same port region.
- If you remove the 802.1Q tag-type from a port, the device automatically removes the 802.1Q tag-type from all ports within the same port region.
- The device supports on configured tag-type per device, along with the default tag-type of 8100. For example, if you configure an 802.1Q tag of 9100 on ports 1 – 8, then later configure an 802.1Q tag of 5100 on port 9, the device automatically applies the 5100 tag to all ports in the same port region as port 9, and also changes the 802.1Q tag-type on ports 1 – 8 to 5100.

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 27, 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 27, enter commands such as the following on the untagged edge links of devices C and D.

```
BigIron RX(config)# tag-type 9100 e3/1 to 3/2
BigIron RX(config)# aggregated-vlan
```

Note that since ports 11 and 12 belong to the port region 1 – 12, the 802.1Q tag actually applies to ports 1 – 12.

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.

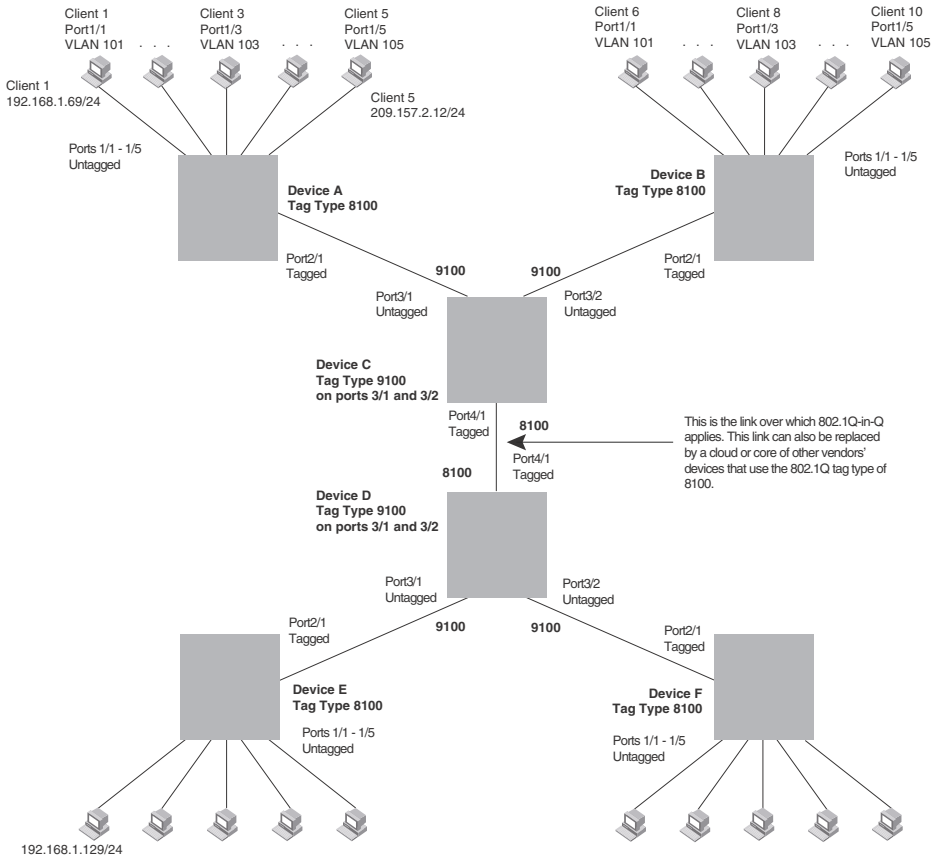
The **ethernet** <*port number*> *to* <*port number*> parameter specifies the ports that will use the defined 802.1Q tag. This parameter operates with the following rules:

- If you specify a single port number, the 802.1Q tag applies only to that port. You can use the **show running-config** command to view how the command has been applied.
- 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 27 shows an example 802.1Q-in-Q configuration.

FIGURE 27 Example 802.1Q-in-Q configuration



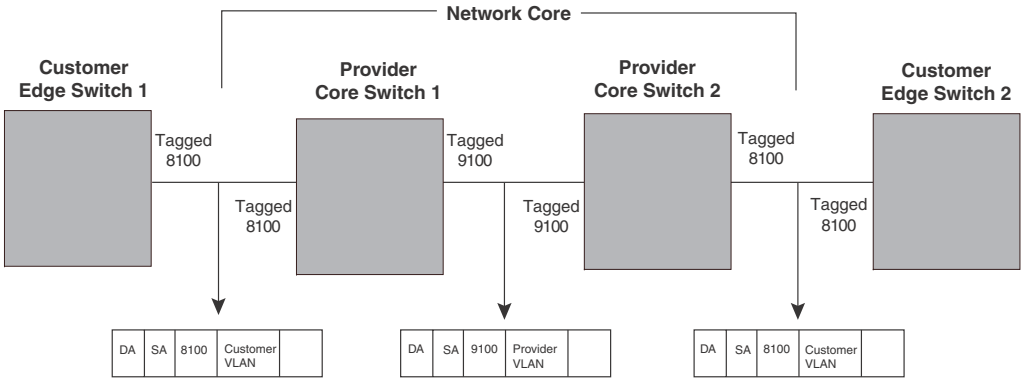
Configuring 802.1q tag-type translation

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-types per port group. This enhancement allows for tag-type translation from one port group to the next on tagged interfaces.

802.1Q tag-type translation enables you to configure 802.1q tag-types per port group, allowing for tag-type translation from one port group to the next on tagged interfaces.

Figure 28 shows a basic example application of the 802.1q tag-type translation feature.

FIGURE 28 802.1q tag-type translation configuration example 1



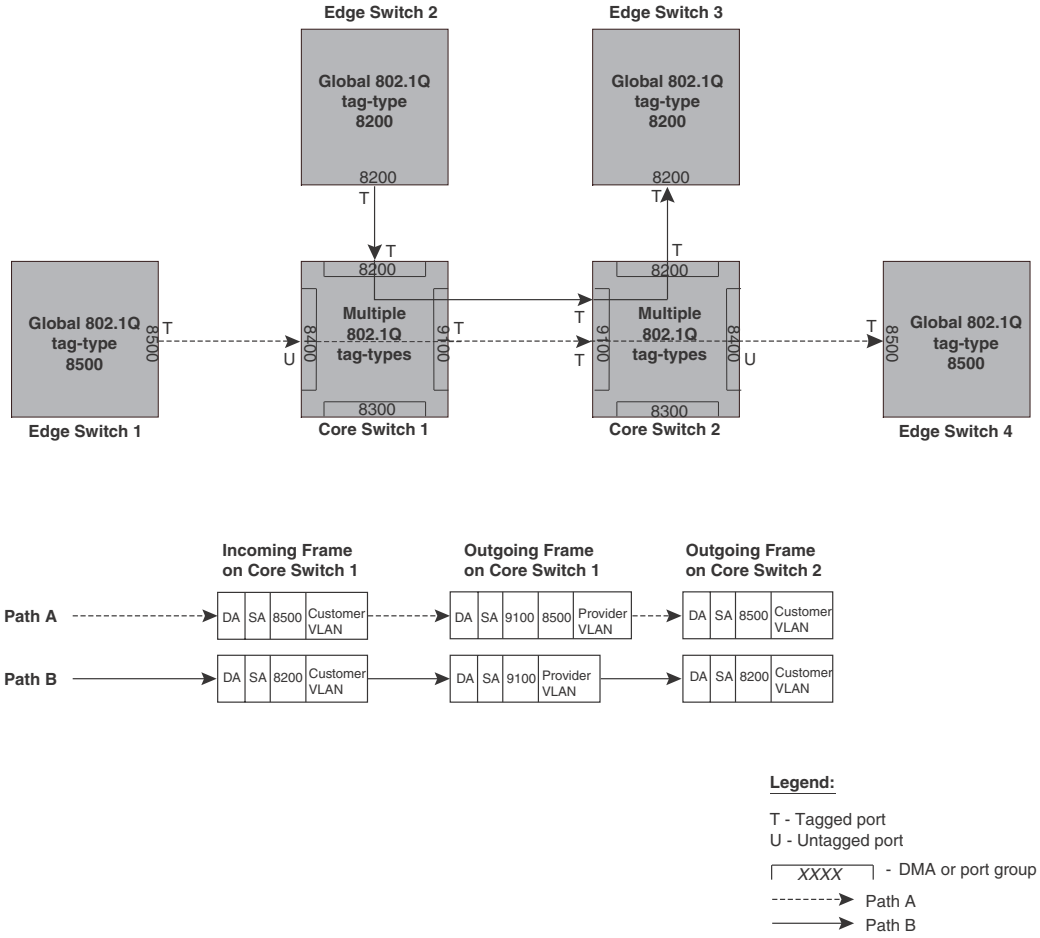
As illustrated in Figure 28, the devices process the packet as follows:

- 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 28 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 29 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 29 802.1q tag-type translation configuration example 2



As illustrated in Figure 29, 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).

For more information, refer to "Configuring 802.1q tag-type translation" on page 302.

Configuration rules

- On the supported devices, you configure 802.1q tag-types per port region. Use the **show running-config** command at any level of the CLI to view port regions. Note that on Gigabit Ethernet modules, ports 1 and 2 belong to the same port region.

- Since the uplink (to the provider cloud) and the edge link (to the customer port) must have different 802.1q tag-types, make sure the uplink and edge link are in different port regions.
- If you configure a port with an 802.1q tag-type, the device automatically applies the 802.1q tag-type to all ports within the same port region.
- If you remove the 802.1q tag-type from a port, the device automatically removes the 802.1q tag-type from all ports within the same port region.
- *Brocade* does not recommend configuring different 802.1q tag-types on ports that are part of a multi-slot trunk. Use the same 802.1q tag-type for all ports in a multi-slot trunk.
- Multiple 802.1Q tag types can be assigned to an interface module. Depending on the module, an 802.1Q tag can be assigned to an individual port or to a group of ports. Table 63 describes the granularity at which each of the device interface modules can have 802.1Q tag-types assigned.

TABLE 63 802.1Q tag-type assignments by module

module type	802.1Q tag-type assignment
4 x 10G	per port
24 x 1G	per 12 ports: 1 - 12, 13 - 24,

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 28). Enter commands such as the following.

```
BigIron RX(config)# tag-type 9100 e 11 to 12
BigIron RX(config)# aggregated-vlan
```

Note that since ports 11 and 12 belong to the port region 9 – 16, the 802.1q tag-type actually applies to ports 9 – 16.

NOTE
Do not configure 802.1q tag-type translation on the edge link (to the customer edge switch).

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. Note that you must specify a value other than 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:

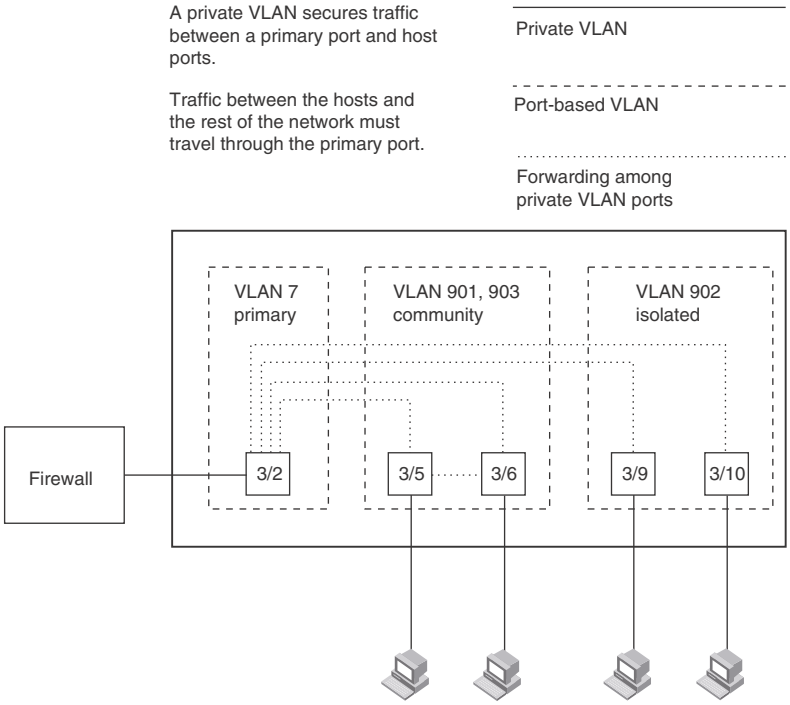
- If you specify a single port number, the 802.1q tag-type applies to all ports within the port region. For example, if you enter the command **tag-type 9100 e 1**, the device automatically applies the 802.1q tag to ports 1 – 8 since all of these ports are in the same port region (controlled by the same DMA). Use the **show running-config** command at any level of the CLI to view port regions. Note that on Gigabit Ethernet modules, ports 1 and 2 belong to the same port region.
- If the port that you specify is part of a multi-slot trunk, the device automatically applies the 802.1q tag-type to all of the ports that are part of the multi-slot trunk.

- If you do not specify a port or range of ports, the 802.1q tag-type applies to all Ethernet ports on the device.

Private VLANs

A private VLAN is a VLAN that has the properties of standard Layer 2 port-based VLANs but also provides additional control over flooding packets on a VLAN. Figure 30 shows an example of an application using a private VLAN.

FIGURE 30 Private VLAN used to secure communication between a workstation and servers



This example uses a private VLAN to secure traffic between hosts and the rest of the network through a firewall. Five ports in this example are members of a private VLAN. The first port (port 3/2) is attached to a firewall. The next four ports (ports 3/5, 3/6, 3/9, and 3/10) are attached to hosts that rely on the firewall to secure traffic between the hosts and the rest of the network. In this example, two of the hosts (on ports 3/5 and 3/6) are in a community private VLAN, and thus can communicate with one another as well as through the firewall. The other two hosts (on ports 3/9 and 3/10), are in an isolated VLAN and thus can communicate only through the firewall. The two hosts are secured from communicating with one another even though they are in the same VLAN.

By default, the private VLAN does not forward broadcast or unknown-unicast packets from outside sources into the private VLAN. If needed, you can override this behavior for broadcast packets, unknown-unicast packets, or both. (Refer to "Enabling broadcast, multicast or unknown unicast traffic to the private VLAN" on page 310.)

You can configure a combination of the following types of private VLANs:

- **Primary** – The primary private VLAN ports are "promiscuous". They can communicate with all the isolated private VLAN ports and community private VLAN ports in the isolated and community VLANs that are mapped to the promiscuous port.

- **Secondary** – The secondary private VLAN are secure VLANs that are separated from the rest of the network by the primary private VLAN. Every secondary private VLAN needs to be associated with a primary private VLAN. There are 2 different types of secondary private VLANs - 'community' and 'isolated' private VLANs:
  - **Isolated** – Broadcasts and unknown unicasts received on isolated ports are sent only to the primary port. They are not flooded to other ports in the isolated VLAN.
  - **Community** – Broadcasts and unknown unicasts received on community ports are sent to the primary port and also are flooded to the other ports in the community VLAN.

Each private VLAN must have a primary VLAN. The primary VLAN is the interface between the secured ports and the rest of the network. The private VLAN can have any combination of community and isolated VLANs. (Refer to "Configuration rules" on page 308.)

Table 64 list the differences between private VLANs and standard VLANs.

TABLE 64 Comparison of private VLANs and standard port-based VLANs

Forwarding behavior	Private VLANs	Standard VLANs
All ports within a VLAN constitute a common Layer broadcast domain	No	Yes
Broadcasts and unknown unicasts are forwarded to all the VLAN's ports by default	No (isolated VLAN) Yes (community VLAN)	Yes
Known unicasts	Yes	Yes

Implementation notes

- The private VLAN implementation in the current release uses the CPU for forwarding packets on the primary VLAN's "promiscuous" port. Other forwarding is performed in the hardware. Support for the hardware forwarding in this feature sometimes results in multiple MAC address entries for the same MAC address in the device's MAC address table. In this case, each of the entries is associated with a different VLAN. The multiple entries are a normal aspect of the implementation of this feature and do not indicate a software problem.
- By default, the primary VLAN does not forward broadcast or unknown unicast packets into the private VLAN. You also can use MAC address filters to control traffic forwarded into and out of the private VLAN. If you are implementing the private VLAN on a Layer 2 Switch, you also can use ACLs to control the traffic into and out of the private VLAN.

Configuration notes

- When Private VLAN mappings are enabled, the device forwards unknown unicast, unknown multicast, and broadcast packets in software. By default, the device forwards unknown unicast, unknown multicast, and broadcast packets in hardware.
- Release 02.4.00 supports private VLANs on untagged ports only. You cannot configure isolated, community, or primary VLANs on 802.1Q tagged ports.

- The BigIron RX forwards all known unicast traffic in hardware. This differs from the way the BigIron implements private VLANs, in that the BigIron uses the CPU to forward packets on the primary VLAN's "promiscuous" port. In addition, on the BigIron, support for the hardware forwarding in this feature sometimes results in multiple MAC address entries for the same MAC address in the device's MAC address table. On the device, multiple MAC entries do not appear in the MAC address table because the BigIron RX transparently manages multiple MAC entries in hardware.
- There is currently no support for IGMP Snooping within Private VLANs. In order to let clients in Private VLANs get multicast traffic, IGMP Snooping must be disabled, so that all multicast packets are treated as unregistered multicast packets and get flooded in software to all the ports.
- You can configure private VLANs and dual-mode VLAN ports on the same device. However, the dual-mode VLAN ports cannot be members of Private VLANs.
- A primary VLAN can have multiple ports. All these ports are active, but the ports that will be used depends on the private VLAN mappings. Also, secondary VLANs (isolated and community VLANs) can be mapped to multiple primary VLAN ports. For example:

```
pvlan mapping 901 ethernet 1/2
pvlan mapping 901 ethernet 2/2
pvlan mapping 901 ethernet 3/2
```

Configuring a private VLAN

To configure a private VLAN, configure each of the component VLANs (isolated, community, and public) as a separate port-based VLAN:

- Use standard VLAN configuration commands to create the VLAN and add ports.
- Identify the type private VLAN type (isolated, community, or public)
- For the primary VLAN, map the other private VLANs to the ports in the primary VLAN

Configuration rules

NOTE
Although a private VLAN resides within a port-based VLAN, the VLAN is considered to be exclusively a private VLAN, not a port-based VLAN.

- You cannot use the private VLAN feature and the dual-mode VLAN port feature on the same device.
- The Spanning Tree Protocol (STP) is independent of this feature, and can be enabled or disabled in the individual port-based VLANs. However, private VLANs are not supported with single-instance STP ("single span").
- You can configure only one private VLAN within a given port-based VLAN. Thus, you must configure a separate port-based VLAN for each private VLAN.
- Each private VLAN can have only one primary VLAN and can not belong LACP ports.
- Each private VLAN can have multiple isolated or community VLANs. You can use any combination of isolated or community VLANs with the primary VLAN. You do not need to use both isolated and community VLANs in the private VLAN.

- You can configure the primary VLAN before or after you configure the community or isolated VLANs. You are not required to configure a specific type of private VLAN before you can configure the other types.
- The ports in all three types of private VLANs can be untagged.
- The primary VLAN has only one active port. The primary VLAN can have more than one port, but only the lowest-numbered available port is active. The other ports provide redundancy.
- You cannot configure the default VLAN (VLAN 1) as a private VLAN.

Configuring an isolated or community private VLAN

To configure an isolated or a community private VLAN, use the following CLI methods.

Using the CLI

To configure a community private VLAN, enter commands such as the following.

```
BigIron RX(config)# vlan 901
BigIron RX(config-vlan-901)# untagged ethernet 3/5 to 3/6
BigIron RX(config-vlan-901)# pvlan type community
```

These commands create port-based VLAN 901, add ports 3/5 and 3/6 to the VLAN as untagged ports, then specify that the VLAN is a community private VLAN.

Syntax: untagged ethernet [to <*portnum*> | ethernet <*portnum*>]

Syntax: [no] pvlan type community | isolated | primary

The **untagged** command adds the ports to the VLAN.

The **pvlan type** command specifies that this port-based VLAN is a private VLAN.

- **community** – Broadcasts and unknown unicasts received on community ports are sent to the primary port and also are flooded to the other ports in the community VLAN.
- **isolated** – Broadcasts and unknown unicasts received on isolated ports are sent only to the primary port. They are not flooded to other ports in the isolated VLAN.
- **primary** – The primary private VLAN ports are “promiscuous”. They can communicate with all the isolated private VLAN ports and community private VLAN ports in the isolated and community VLANs that are mapped to the promiscuous port.

Configuring the primary VLAN

Use the following CLI method to configure the primary VLAN.

Using the CLI

To configure a primary private VLAN, enter commands such as the following.

BigIron RX(config)# vlan 7
BigIron RX(config-vlan-7)# untagged ethernet 3/2
BigIron RX(config-vlan-7)# pvlan type primary
BigIron RX(config-vlan-7)# pvlan mapping 901 ethernet 3/2

These commands create port-based VLAN 7, add port 3/2 as an untagged port, identify the VLAN as the primary VLAN in a private VLAN, and map the other private VLANs to the ports in this VLAN.

Syntax: untagged ethernet <*portnum*> [to <*portnum*> | ethernet <*portnum*>]

Syntax: [no] pvlan type community | isolated | primary

Syntax: [no] pvlan mapping <vlan-id> ethernet <portnum>

The **untagged** command adds the ports to the VLAN.

The **pvlan type** command specifies that this port-based VLAN is a private VLAN. Specify **primary** as the type.

The **pvlan mapping** command identifies the other private VLANs for which this VLAN is the primary. The command also specifies the primary VLAN ports to which you are mapping the other private VLANs.

- The <*vlan-id*> parameter specifies another private VLAN. The other private VLAN you want to specify must already be configured.
- The **ethernet** <*portnum*> parameter specifies the primary VLAN port to which you are mapping all the ports in the other private VLAN (the one specified by <*vlan-id*>).

Enabling broadcast, multicast or unknown unicast traffic to the private VLAN

To enhance private VLAN security, the primary private VLAN does not forward broadcast or unknown unicast packets to its community and isolated VLANs. For example, if port 3/2 in Figure 30 on page 306 receives a broadcast packet from the firewall, the port does not forward the packet to the other private VLAN ports (3/5, 3/6, 3/9, and 3/10).

This forwarding restriction does not apply to traffic from the private VLAN. The primary port does forward broadcast and unknown unicast packets that are received from the isolated and community VLANs. For example, if the host on port 3/9 sends an unknown unicast packet, port 3/2 forwards the packet to the firewall.

If you want to remove the forwarding restriction, you can enable the primary port to forward broadcast or unknown unicast traffic, if desired, using the following CLI method. You can enable or disable forwarding of broadcast or unknown unicast packets separately.

Using the CLI

To configure the ports in the primary VLAN to forward broadcast, multicast or unknown unicast traffic received from sources outside the private VLAN, enter the following commands at the global CONFIG level of the CLI.

```
BigIron RX(config)# pvlan-preference broadcast flood
BigIron RX(config)# pvlan-preference unknown-unicast flood
```

These commands enable forwarding of broadcast, multicast and unknown-unicast packets to ports within the private VLAN. To again disable forwarding, enter a command such as the following.

```
BigIron RX(config)# no pvlan-preference broadcast flood
```

This command disables forwarding of broadcast packets within the private VLAN.

Syntax: [no] pvlan-preference broadcast | unknown-unicast flood

CLI example for Figure 30

To configure the private VLANs shown in Figure 30 on page 306, enter the following commands.

```
BigIron RX(config)# vlan 901
BigIron RX(config-vlan-901)# untagged ethernet 3/5 to 3/6
BigIron RX(config-vlan-901)# pvlan type community
BigIron RX(config-vlan-901)# exit
BigIron RX(config)# vlan 902
BigIron RX(config-vlan-902)# untagged ethernet 3/9 to 3/10
BigIron RX(config-vlan-902)# pvlan type isolated
BigIron RX(config-vlan-902)# exit
BigIron RX(config)# vlan 903
BigIron RX(config-vlan-903)# untagged ethernet 3/5 to 3/6
BigIron RX(config-vlan-903)# pvlan type community
BigIron RX(config-vlan-903)# exit
BigIron RX(config)# vlan 7
BigIron RX(config-vlan-7)# untagged ethernet 3/2
BigIron RX(config-vlan-7)# pvlan type primary
BigIron RX(config-vlan-7)# pvlan mapping 901 ethernet 3/2
BigIron RX(config-vlan-7)# pvlan mapping 902 ethernet 3/2
BigIron RX(config-vlan-7)# pvlan mapping 903 ethernet 3/2
```

Other 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 device. Although this is the default maximum, the device can support up to 4089 VLANs and 4095 virtual routing interfaces. (VLAN IDs 0, 4090, 4091, 4092 and 4095 are reserved.)

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.

```
BigIron RX(config)# system-max vlan 2048
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# reload
```

Syntax: [no] system-max vlan <*num*>

The <*num*> parameter specifies the maximum number of VLANs that can be configured. Enter 1 – 4089, but IDs 4091 – 4094 are reserved.

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 needed to be sent on all ports of the VLAN (or "flooded" across the VLAN).

By default, the device performs *hardware* flooding for Layer 2 multicast and broadcast packets. (Layer 2 multicast packets have a multicast address in the destination MAC address field.) However, if uplink VLANs or protocol-based VLANs are configured, this default behavior is overridden and *software* flooding is enabled.

You can disable hardware flooding for Layer 2 multicast and broadcast packets on a per-VLAN basis. For example:

```
BigIron RX(config)#
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# no multicast-flooding
```

Syntax: [no] multicast-flooding

NOTES:

- This feature is supported on the 10 Gigabit Ethernet module.
- 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.
- This feature is not supported on private VLANs.
- You cannot enable this feature on the designated management VLAN for the device.
- If you enable this feature on a VLAN that includes a trunk group, hardware flooding for Layer 2 multicast and broadcast packets occurs only on the trunk group's primary port. Multicast and broadcast traffic for the other ports in the trunk 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).

By default, the device performs *hardware* flooding for unknown unicast packets. However, if uplink VLANs or protocol-based VLANs are configured, this default behavior is overridden and *software* flooding is enabled.

To disable unicast hardware flooding on a VLAN ports and enable software flooding, enter commands such as the following.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# no unknown-unicast-flooding
BigIron RX(config-vlan-2)# exit
BigIron RX(config)# reload
```

Syntax: [no] unknown-unicast-flooding

Flow based MAC learning

In this release, the **cpu-flooding** command that disables hardware flooding of unknown unicast, multicast, and broadcast packets on all VLAN has been added. When using this command, unknown unicast packets will go to the CPU and will be CPU forwarded. Source MAC learning will be done by CPU. The first packet for unknown DA will go to the CPU and CPU will program the hardware on demand. Any supplemental packets will be forwarded by the hardware.This will allow MAC learning only where necessary and at a system level to allow more than 16k MACs. In addition, a MAC will be learned only on the port or packet processor it was received.

Enabling CPU flooding

To enable CPU based flooding of unknown unicast, broadcast and multicast packets, enter the following command at the global configuration level.

```
BigIron RX(config)# cpu-flooding
```

To enable flow based MAC learning and CPU flooding for unknown unicast packets only, enter the following command at the global configuration level.

```
BigIron RX(config)# cpu-flooding unknown-unicast
```

To enable CPU based flooding for broadcast and multicast packets, enter the following command at the global configuration level.

BigIron RX(config)# cpu-flooding multicast

Syntax: [no] cpu-flooding [multicast] [unknown-unicast]

Use the **multicast** parameter to specify CPU flooding for broadcast and multicast packets.

Use the **unknown-unicast** parameter to specify CPU flooding for unknown unicast packets only.

NOTE
This command does not erase any multicast or unknown-unicast flooding configuration. If this command is enabled, then it supersedes the per-vlan configuration.

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.

```
BigIron RX(config)# vlan 10 by port
BigIron RX(config-vlan-10)# untag ethernet 1/1 to 1/24
BigIron RX(config-vlan-10)# untag ethernet 2/1 to 2/2
BigIron RX(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, 24 ports on a 10/100 module and two Gigabit ports on a Gigabit module are added to port-based VLAN 10. The two Gigabit ports are then configured as uplink ports.

Configuring control protocols in VLANs

You can configure the following protocols on a VLAN:

- MRP (Refer to Chapter 14, "Metro Ring Protocol (MRP) Phase 1 and 2".)
- VSRP (Refer to Chapter 15, "Virtual Switch Redundancy Protocol (VSRP)".)
- STP (Refer to Chapter 12, "Configuring Spanning Tree Protocol".)
- RSTP (Refer to Chapter 13, "Configuring Rapid Spanning Tree Protocol".)

Other configuration options

You can also configure the following on a VLAN:

- "Configuring static ARP entries" on page 132
- "Setting maximum frame size per PPCR" on page 172

Displaying VLAN information

After you configure the VLANs, you can view and verify the configuration.

Displaying VLAN information

Enter the following command at any CLI level.

```
BigIron RX# show vlan
Configured PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 4095
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
L2 protocols   : NONE
Untagged Ports : ethe 2/1 to 2/24 ethe 3/1 to 3/24 eth
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   : ethe 3/1 ethe 3/12 to 3/13 ethe 3/24
```

Syntax: show vlan [<*vlan-id*>] [| [begin <*expression*> | exclude <*expression*> | include <*expression*>]

Enter a VLAN ID if you want to display information for a specific VLAN.

The output shows the following information.

TABLE 65 Output of show vlan

This field...	Displays...
Configured PORT-VLAN entries	Number of port-based VLANs in the configuration.
Maximum PORT-VLAN entries: 4095	Maximum number of port-based VLANs that you can configure. Note however, IDs 4091 and 4092 are reserved for control purposes.
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/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.

Displaying VLAN information for specific ports

To determine which VLANs a port is a member of, enter the following command.

```
BigIron RX# 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 66 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.

```
BigIron RX# show vlan detail
Untagged Ports  : ethe 2/1 to 2/24 ethe 4/4
Tagged Ports    : None
Dual-mode Ports : ethe 3/1 to 3/24 ethe 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 67 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 >

TABLE 67 Output of show vlan detail (Continued)

This field...	Displays...
Type	Port type: physical or trunk
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.

```
BigIron RX# 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: ethe 3/1
```

Syntax: show vlan-group [vlan-group-id] [| [begin <*expression*> | exclude <*expression*> | include <*expression*>]

The output shows the following information.

TABLE 68 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

Transparent firewall mode

TheTransparent Firewall mode allows the device to switch self-originated control packets. By default, *Brocade* devices will drop control packets received with the device's MAC address as the packet's source MAC address (i.e. self originated packet from the switch or router). Under the Transparent Firewall mode, switching of self-originated packets is allowed. The Transparent Firewall mode feature is a per VLAN configuration and is disabled by default.

Enabling a transparent firewall

To set the mode to transparent, enter a command such as the following.

```
BigIron RX(config-vlan-10)# transparent-fw-mode
```

To set the mode to routed, enter a command such as the following.

```
BigIron RX(config-vlan-10)# no transparent-fw-mode
```

Syntax: [no] transparent-fw-mode

Chapter

Configuring Spanning Tree Protocol 12

In this chapter

IEEE 802.1D Spanning Tree Protocol (STP)

The device 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.

NOTE
The total number of supported STP, RSTP, or MSTP indices is 128.

Enabling or disabling STP

STP is disabled by default on the device. Thus, new VLANs you configure on the device have STP disabled by default. Table 69 lists the default STP states for the device.

TABLE 69 Default STP states

Device type	Default STP type	Default STP state	Default STP state of new VLANs
device	*Brocade*'s multiple instances of spanning tree	Disabled	Disabled

By default, each VLAN on a BigIron RX runs a separate spanning tree instance. Each device has one VLAN (VLAN 1) by default that contains all of its ports. However, if you configure additional port-based VLANs on a device, 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 device.

- **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 trunk group, the change affects all ports in the trunk group.

Enabling or disabling STP globally

Use the following methods to enable or disable STP on the device 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.

To enable STP for all ports in all VLANs on a device, enter the following command.

```
BigIron RX(config)# spanning-tree
```

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 device 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.

```
BigIron RX(config)# vlan 10
BigIron RX(config-vlan-10)# spanning-tree
```

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 trunk group, the change affects all ports in the trunk group.

To enable STP on an individual port, enter commands such as the following.

```
BigIron RX(config)# interface 1/1
BigIron RX(config-if-e1000-1/1)# spanning-tree
```

Syntax: [no] spanning-tree

Default STP bridge and port parameters

Table 70 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.

TABLE 70 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, *Brocade* recommends 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 71 lists the default STP port parameters. The port parameters affect individual ports and are separately configurable on each port.

TABLE 71 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 BigIron RX's STP bridge priority to the highest value, so as to make the device the root bridge, enter the following command.

```
BigIron RX(config)# vlan 20
BigIron RX(config-vlan-20)# spanning-tree priority 0
```

To make this change in the default VLAN, enter the following commands.

```
BigIron RX(config)# vlan 1
BigIron RX(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 70 on page 321.

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.

```
BigIron RX(config)# vlan 10
BigIron RX(config-vlan-10)# spanning-tree ethernet 1/5 path-cost 15 priority 64
```

Syntax: 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 71 on page 321. 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.

STP root guard

In release 02.3.00, a new security feature that allows a port to run STP but not allow the connected device to become the Root has been added. The STP Root Guard feature provides a way to enforce the root bridge placement in the network and trigger errors if any changes from the root bridge placement are detected. This feature allows STP to interoperate with user network bridges while still maintaining the bridged network topology that the administrator requires.

When Root Guard is enabled on a port, it keeps the port in designated FORWARDING state. If the port receives a superior STP BPDU, it sets the port into BLOCKING and triggers a log 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 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 protect 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.

Enabling STP root guard

A STP Root Guard is configured on a per interfaces basis. To enable a Root Guard, enter a command such as the following.

```
BigIron RX (config)#interface ethernet 5/5
BigIron RX(config-if-e10000-5/5)spanning-tree root-protect
```

Syntax: [no] spanning-tree root-protect

Enter the no form of the command to disable STP Root Guard on the port.

Setting the STP root guard timeout period

To configure the STP Root protect timeout period globally, enter a command such as the following.

```
BigIron RX(config)# spanning-tree root-protect timeout 120
```

Syntax: 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.

Displaying the STP root guard

To display the STP Root Guard state, enter the **show spanning-tree root-protect** command.

```
BigIron RX#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

Sample Syslog messages

A Syslog message such as the following is generated after the Root Guard blocks a port.

```
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.

```
STP: Root Guard Port 12/21, VLAN 10 consistent (Timeout)
```

Spanning Tree Protocol (STP) BPDU guard

STP protection provides the ability to prohibit an end station from initiating or participating in an STP topology. The STP BPDU Guard is used to keep all active network topologies predictable.

The spanning-tree protocol 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 Bridge Protocol Data Units (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 STP BPDU, triggering an STP topology change.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU Guard feature on the *Brocade* port to which the end station is connected. *Brocade*'s STP BPDU Guard feature disables the connected device's ability to initiate or participate in an STP topology change, by dropping all BPDUs received from the connected device.

Enabling STP protection

You can enable STP BPDU Guard on a per-port basis.

To prevent an end station from initiating or participating in STP topology changes, enter the following command at the interface level of the CLI.

```
BigIron RX(config) interface ethe 2/1
BigIron RX(config-if-e1000-2/1)# spanning-tree protect
```

This command causes the port to drop STP BPDUs sent from the device on the other end of the link.

Syntax: [no] spanning-tree protect

Enter the **no** form of the command to disable STP protection on the port.

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.

```
BigIron RX# 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 device's Per VLAN Spanning Tree (PVST+) compatibility configuration. Refer to "PVST/PVST+ compatibility" on page 332.

The <*num*> parameter displays only the entries after the number you specify. For example, on a device 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 328.

The **show spanning-tree** command shows the following information.

TABLE 72 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 72 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	**NOTE:** If you configure this value, specify it in decimal format. Refer to “Changing STP port parameters” on page 322.
Path Cost	The port’s STP path cost.

TABLE 72 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.

```
BigIron RX# show spanning-tree detail vlan 10
VLAN 10 - STP instance 1
------------------------------------------------------------------
STP Bridge Parameters:

Bridge identifier - 0x8000000480a04000
Root bridge - 0x8000000480a04000
Control ports - ethe 1/3 ethe 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 - ethe 1/3 ethe 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 device 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 73 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: • *Brocade* 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.

TABLE 73 CLI display of detailed STP information for ports (Continued)

This field...	Displays...
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 device runs a separate spanning tree, which you can enable or disable on an individual VLAN basis.

Alternatively, you can configure the device to run a single spanning tree across all of its ports and VLANs. The SSTP feature is especially useful for connecting a device 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 device. Refer to "Default STP bridge and port parameters" on page 321.

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 device 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 run MSTP instead. In MSTP, each VLAN has its own spanning tree. VLANs that were not members of the single spanning tree were not enabled for STP. Therefore, STP remains disabled on those VLANs.

Enabling SSTP

NOTE
If the device 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 device contains more than one port-based VLAN and you want all the ports to be in the same STP broadcast domain.

To configure the device to run a single spanning tree, enter the following command at the global CONFIG level.

```
BigIron RX(config)# spanning-tree single
```

NOTE
If the device has only one port-based VLAN, the CLI command for enabling SSTP is not listed in the CLI. The command is listed only if you have configured a port-based VLAN.

To change a global STP parameter, enter a command such as the following at the global CONFIG level.

```
BigIron RX(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.

```
BigIron RX(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 321.

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 376.

Displaying SSTP information

To verify that SSTP is in effect, enter the following commands at any level of the CLI.

```
BigIron RX(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 324.

PVST/PVST+ compatibility

Brocade's support for Cisco's Per VLAN Spanning Tree plus (PVST+) allows the device to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices[1]. *Brocade* ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected.

When it is configured for MSTP, the device can interoperate with PVST.

1. Cisco user documentation for PVST/PVST+ refers to the IEEE 802.1Q spanning tree as the ***Common Spanning Tree (CST)***.

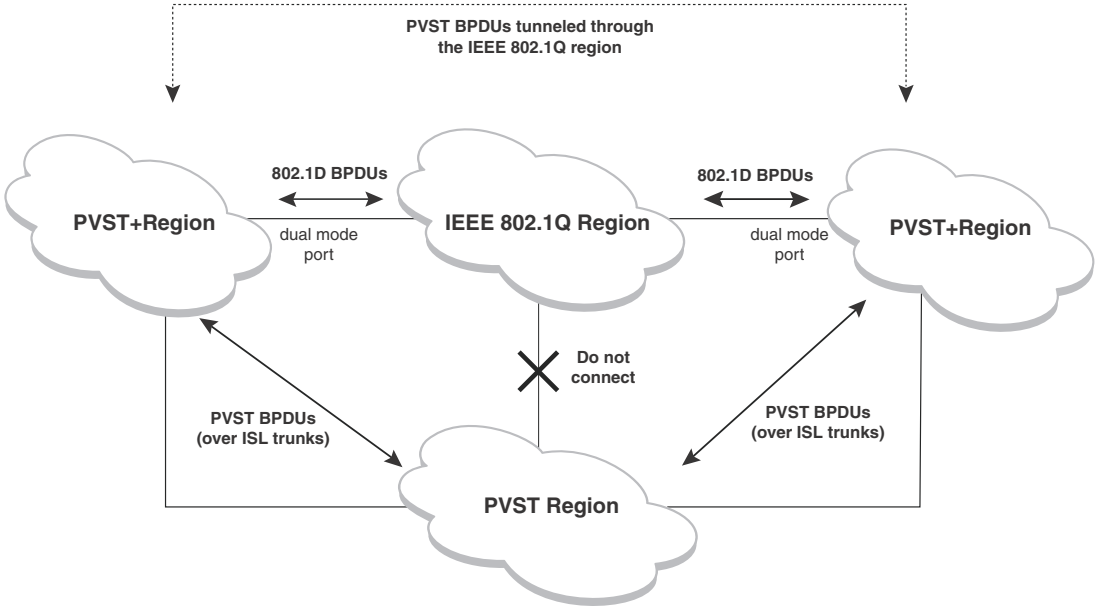
Overview of PVST and PVST+

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 device 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 31 shows the interaction of IEEE 802.1Q, PVST, and PVST+ regions.

FIGURE 31 Interaction of IEEE 802.1Q, PVST, and PVST+ regions



VLAN tags and dual mode

To support the IEEE 802.1Q (Common Spanning Tree) portion of PVST+, a port must be a member of VLAN 1. Cisco devices always use VLAN 1 to support the IEEE 802.1Q portion of PVST+.

For the port to also support the other VLANs (the PVST+ VLANs) in tagged mode. The port must be a dual-mode port.

The untagged frames are supported on the port's *native VLAN*. By default, the native VLAN is the same as the device's *default VLAN*[1], 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.

1. Cisco PVST/PVST+ documentation refers to the Default VLAN as the ***Default Native VLAN***.

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 native VLAN). The Port Native VLAN ID does not need to be the same as the default VLAN.

NOTE
Support for the IEEE 802.1Q spanning tree always uses VLAN 1, regardless of whether the device 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 device.

Enabling PVST+ support manually

To immediately enable PVST+ support on a port, enter commands such as the following.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(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 device, enter the following command at any level of the CLI.

```
BigIron RX(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 74 CLI Display of PVST+ Information

This field...	Displays...
Port	The *Brocade* 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

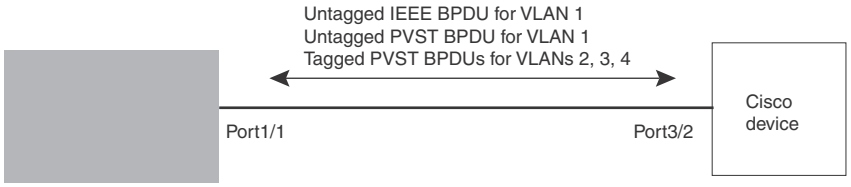
The examples use two common configurations:

- 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 32, a PVST+ configuration uses VLAN 1 as the untagged default VLAN and VLANs 2, 3, and 4 as tagged VLANs.

FIGURE 32 Default VLAN 1 for untagged BPDUs



To implement this configuration, enter the following commands on the RX.

```
BigIron RX(config)# vlan-group 1 vlan 2 to 4
BigIron RX(config-vlan-group-1)# tagged ethernet 1/1
BigIron RX(config-vlan-group-1)# exit
BigIron RX(config)# interface ethernet 1/1
BigIron RX(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.

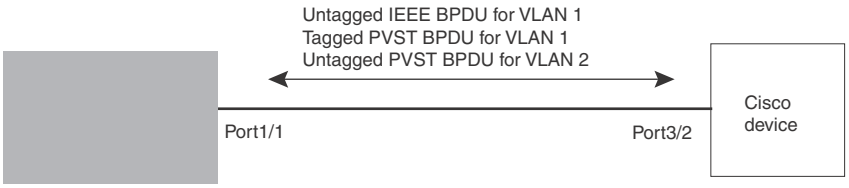
Port 1/1 will process BPDUs as follows:

- 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 33, 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 33 Port native VLAN 2 for untagged BPDUs



To implement this configuration, enter the following commands on the RX.

```
BigIron RX(config)# default-vlan-id 4000
BigIron RX(config)# vlan 1
BigIron RX(config-vlan-1)# tagged ethernet 1/1
BigIron RX(config-vlan-1)# exit
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# untagged ethernet 1/1
BigIron RX(config-vlan-2)# exit
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e10000-1/1)# pvst-mode
BigIron RX(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.

```
BigIron RX(config)# default-vlan-id 1000
BigIron RX(config)# vlan 1
BigIron RX(config-vlan-1)# tagged ethernet 1/1 to 1/2
BigIron RX(config-vlan-1)# exit
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e10000-1/1)# pvst-mode
```

```
BigIron RX(config-if-e10000-1/1)# exit
BigIron RX(config)# interface ethernet 1/2
BigIron RX(config-if-e10000-1/2)# pvst-mode
BigIron RX(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.

```
BigIron RX(config)# default-vlan-id 1000
BigIron RX(config)# vlan 1
BigIron RX(config-vlan-1)# tagged ethernet 1/1 to 1/2
BigIron RX(config-vlan-1)# exit
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e10000-1/1)# pvst-mode
BigIron RX(config-if-e10000-1/1)# exit
BigIron RX(config)# interface ethernet 1/2
BigIron RX(config-if-e10000-1/2)# pvst-mode
BigIron RX(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.

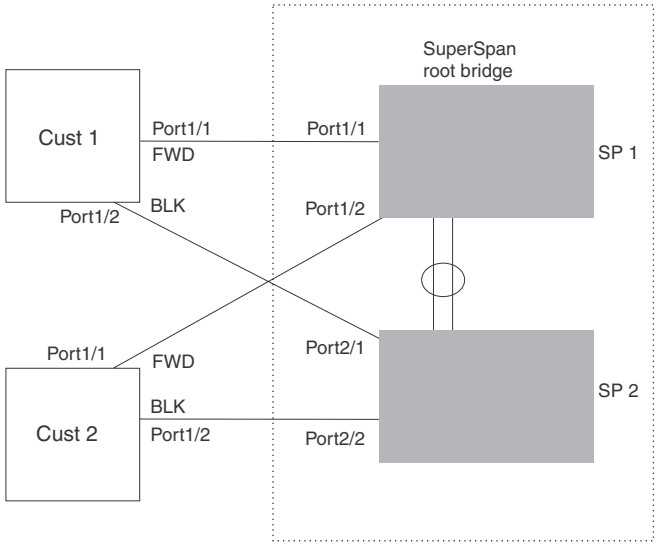
SuperSpan™

SuperSpan is an *Brocade* STP enhancement that allows Service Providers (SPs) to use STP in both SP networks and customer networks. The SP devices are *Brocade* 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 *Brocade* 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 34 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 *Brocade* devices in the SP are running SuperSpan.

FIGURE 34 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 *Brocade* devices in the SP. In Table 34 on page 338, 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 a n *Brocade* device receives a customer's BPDU on a boundary interface, the device 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 *Brocade* device that is configured for SuperSpan forwards the BPDU using the changed destination MAC address. At the other end of the tunnel, the *Brocade* device connected to the customer's network changes the destination MAC address back to the bridge group address (01-80-c2-00-00-00).

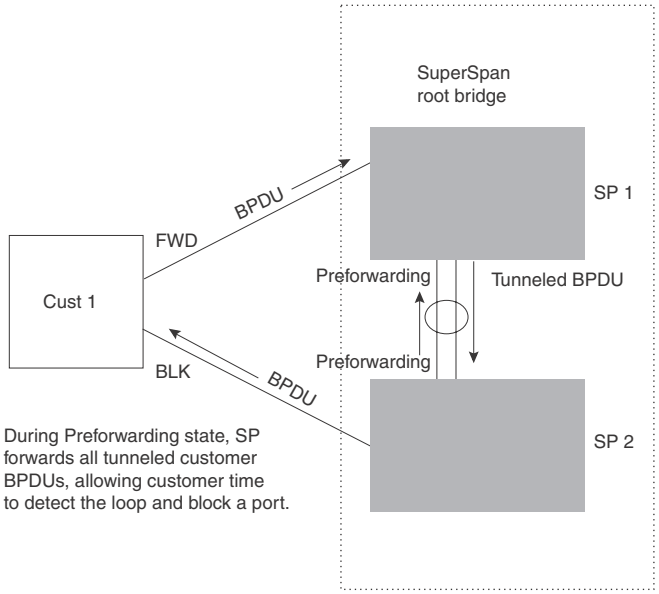
Preforwarding state

To ensure that the customer's network has time to converge at Layer 2 and prevent loops, the *Brocade* 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 *Brocade* device 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 *Brocade* 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 35 shows an example of how the Preforwarding state is used.

FIGURE 35 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 device to a network that is already running SuperSpan, you must enable SuperSpan on the new device, at least on the VLANs that will be tunneling the customer traffic. Otherwise, the new device does not use the Preforwarding state. This can cause temporary loops in the network.

Mixing 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.

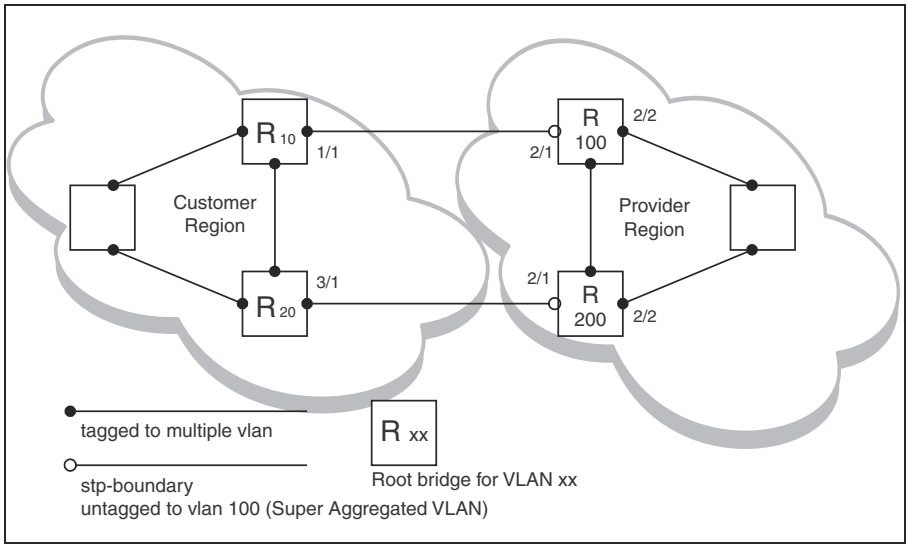
NOTE
All the combinations listed above are supported when the boundary ports joining the SP SuperSpan domain to the client spanning trees are untagged. For example, all these combinations are valid in super aggregated VLAN configurations. If the boundary ports are tagged, you cannot use Single STP in the client network in combination with multiple spanning trees in the SP SuperSpan domain.

The examples below are in super aggregated configuration scenarios.

Customer and SP use multiple spanning trees

Figure 36 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 36 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.

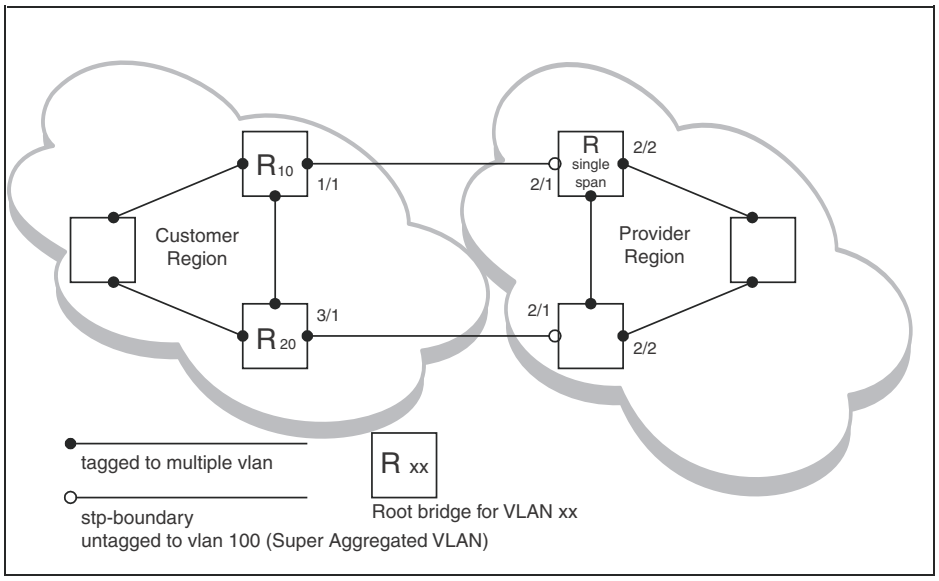
In the above example, STP in VLAN 10 will select R10 as the root bridge and make 1/1 on R10 forwarding while blocking port 3/1 on R20. The opposite occurs for STP in VLAN 20. As a result, both links connecting the customer and SP regions are fully utilized and serve as backup links at the same time, 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 37 shows an example of SuperSpan where the customer network uses multiple spanning trees while the SP network uses Single STP.

FIGURE 37 Customer using Multiple Spanning Trees and SP using single STP



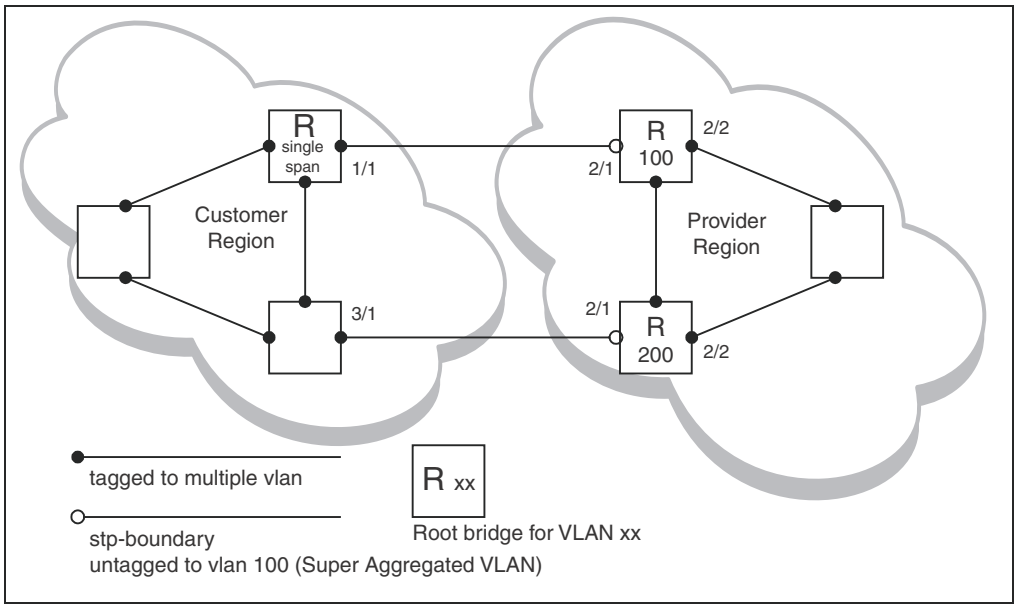
Customer traffic from different VLANs is maintained by different spanning trees, while the SP network is maintained by a single spanning tree. The SP can still use multiple VLANs at the core to separate traffic from different customers. However, all VLANs will have the same network topology because they 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 340, 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 38 shows an example of SuperSpan where the customer network uses Single STP while the SP uses multiple spanning trees.

FIGURE 38 Customer using single STP and SP using Multiple Spanning Trees

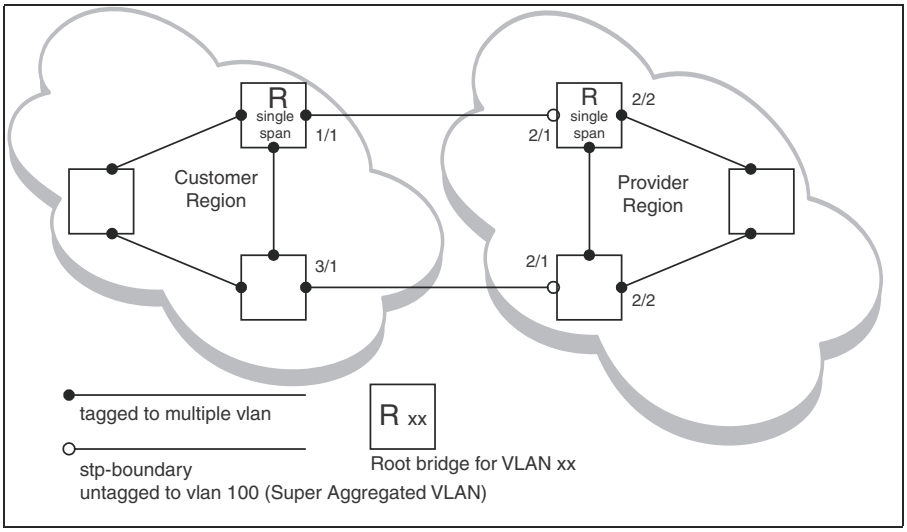


In this setup, the customer network is running a single spanning tree for VLANs 10 and 20. The traffic from VLAN 10 and 20 will be carried, or aggregated by VLAN 100 at the SP's network. The main difference between this scenario and the previous tow scenarios is that all traffic at the customer's network 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 39 shows an example of SuperSpan where the customer network and SP both use Single STP.

FIGURE 39 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 a device for SuperSpan:

- Configure each interface on the device 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 *Brocade* network, the 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 , enter the following commands.

```
BigIron RX(config)# interface 1/1
BigIron RX(config-if-e1000-1/1)# stp-boundary 1
BigIron RX(config)# interface 1/2
BigIron RX(config-if-e1000-1/2)# stp-boundary 2
```

These commands configure two interfaces on the *Brocade* device 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. You can specify a number from 1 – 65535.

To configure the boundary interfaces on SP 2 in Figure 34, enter the following commands.

```
BigIron RX(config)# interface 2/1
BigIron RX(config-if-e1000-2/1)# stp-boundary 1
BigIron RX(config)# interface 2/2
BigIron RX(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 a value from 3 – 30 seconds. The default is 5 seconds.

To globally enable SuperSpan, enter the following command.

```
BigIron RX(config)# super-span-global
```

Syntax: [no] super-span-global [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 device. To disable SuperSpan in an individual VLAN, enter commands such as the following.

```
BigIron RX(config)# vlan 10
BigIron RX(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.

```
BigIron RX(config)# show super-span
CID 1 Boundary Ports:
     Port  C-BPDU  C-BPDU  T-BPDU  T-BPDU
           Rxed    Txed    Rxed    Txed
     1/1   1       0       0       0
     1/2   0       0       0       0
     Total 1       0       0       0

CID 2 Boundary Ports:
     Port  C-BPDU  C-BPDU  T-BPDU  T-BPDU
           Rxed    Txed    Rxed    Txed
     2/1   0       0       3       0
     2/2   0       0       0       0
     Total 0       0       3       0
```

In this example, the device 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 device is shown.

This command shows the following information.

TABLE 75 CLI display of SuperSpan customer ID information

This field...	Displays...
CID	The SuperSpan customer ID number.
Port	The boundary port number.
C-BPDU Rxed	The number of BPDUs received from the client spanning tree.
C-BPDU Txed	The number of BPDUs sent to the client spanning tree.
T-BPDU Rxed	The number of BPDUs received from the SuperSpan tunnel.
T-BPDU Txed	The number of BPDUs sent to the SuperSpan tunnel.

To display general STP information, refer to "Displaying STP information" on page 324.

Chapter

Configuring Rapid Spanning Tree Protocol 13

In this chapter

Overview of Rapid Spanning Tree Protocol

RSTP provides rapid convergence and takes advantage of point-to point wiring of the spanning tree. Failure in one forwarding path does not affect other forwarding paths. RSTP improves the operation of the spanning tree while maintaining backward compatibility.

NOTE
The total number of supported STP, RSTP, or MSTP indices is 128.

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:

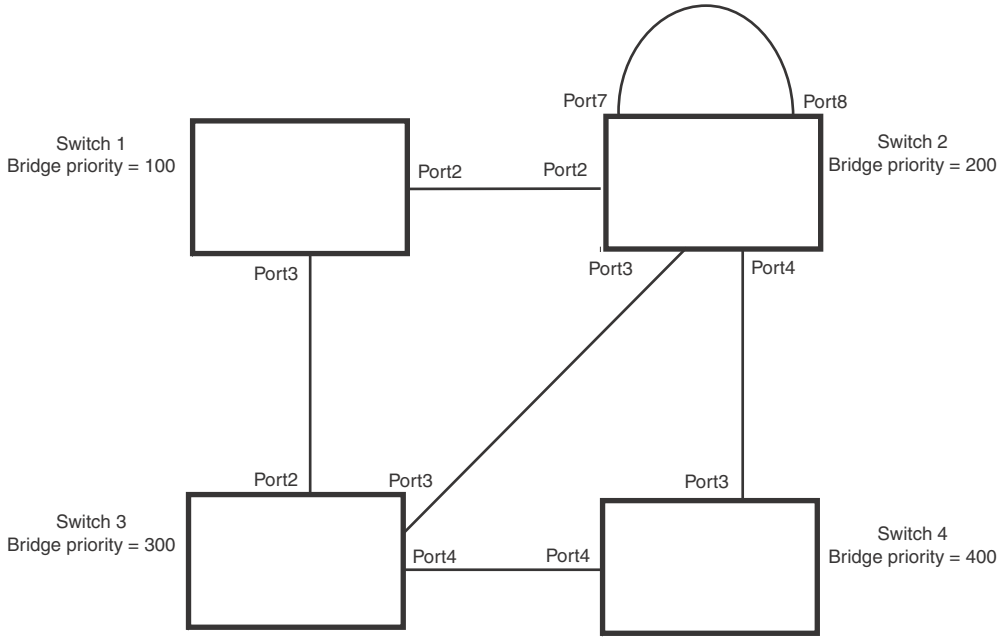
- The port that receives the RST BPDU with the lowest path cost from the root bridge becomes the ***Root port***.
- If two ports on the same bridge are physically connected, the port that receives the superior RST BPDU becomes the ***Backup port***, while the other port becomes the ***Designated port***.
- 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 40) 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 40 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 40 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

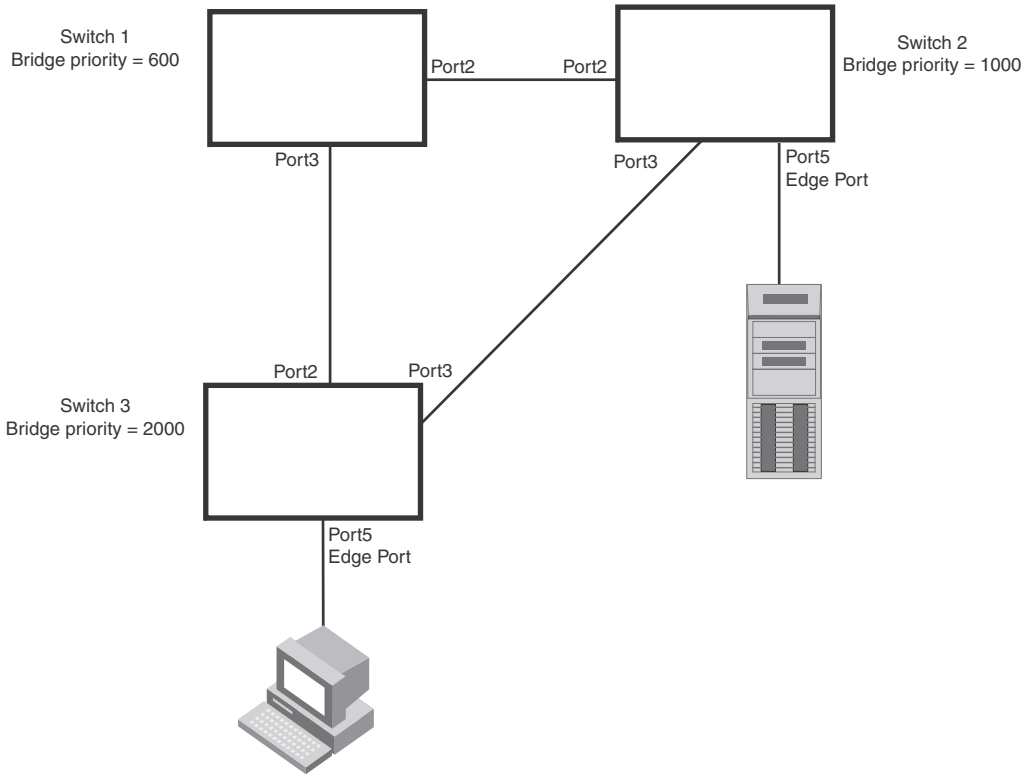
Switch 4 is not directly connected to the root bridge. It has two ports with superior incoming RST BPDUs from two separate LANs: Port3 and Port4. The RST BPDUs received on Port3 are superior to the RST BPDUs received on port 4; therefore, Port3 becomes the Root port and Port4 becomes the Alternate port.

Edge ports and edge port roles

Brocade's implementation of RSTP allows ports that are configured as Edge ports to be present in an RSTP topology. (Figure 41). 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 41 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.

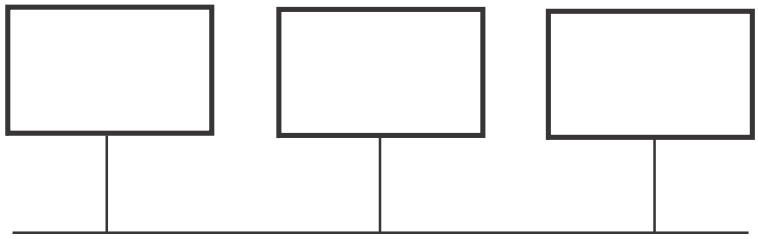
Point-to-point ports

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 42 is an example of shared media that should not be configured as point-to-point links. In Figure 42, a port on a bridge communicates or is connected to at least two ports.

FIGURE 42 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 43, 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:

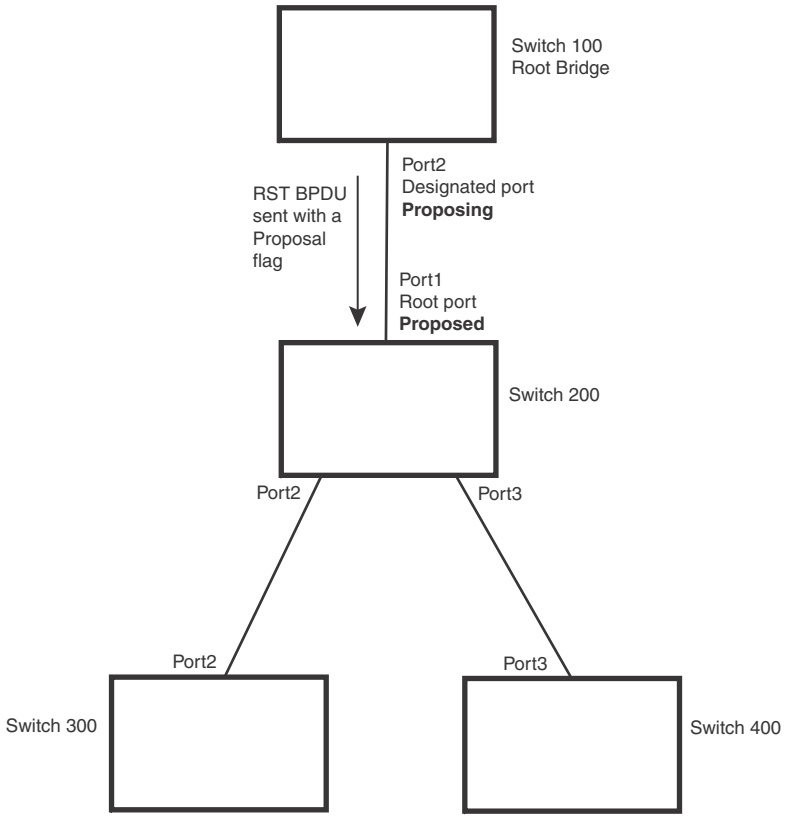
- **Proposing** – The Designated port on the root bridge sends an RST BPDU packet to its peer port that contains a proposal flag. The proposal flag is a signal that indicates that the Designated port is ready to put itself in a forwarding state (Figure 43). The Designated port continues to send this flag in its RST BPDU until it is placed in a forwarding state (Figure 46) or is forced to operate in 802.1D mode. (Refer to "Compatibility of RSTP with 802.1D" on page 374)
- **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 43):

- 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 "Bridges and bridge port roles" on page 347.)
- 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.

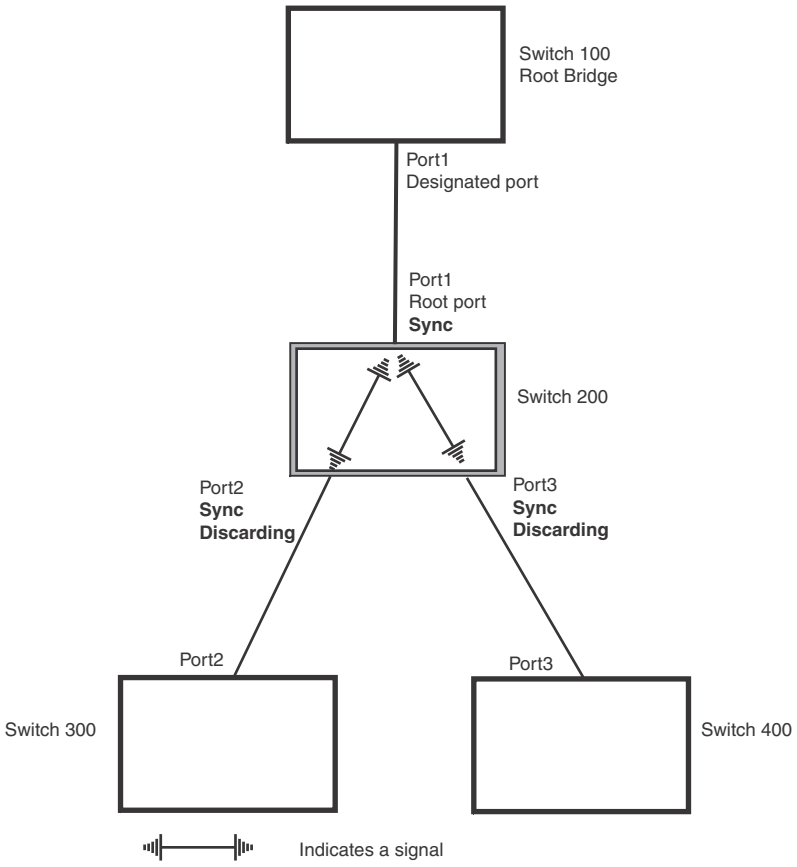
In Figure 43, Port3/Switch 200 is elected as the Root port

FIGURE 43 Proposing and proposed stage



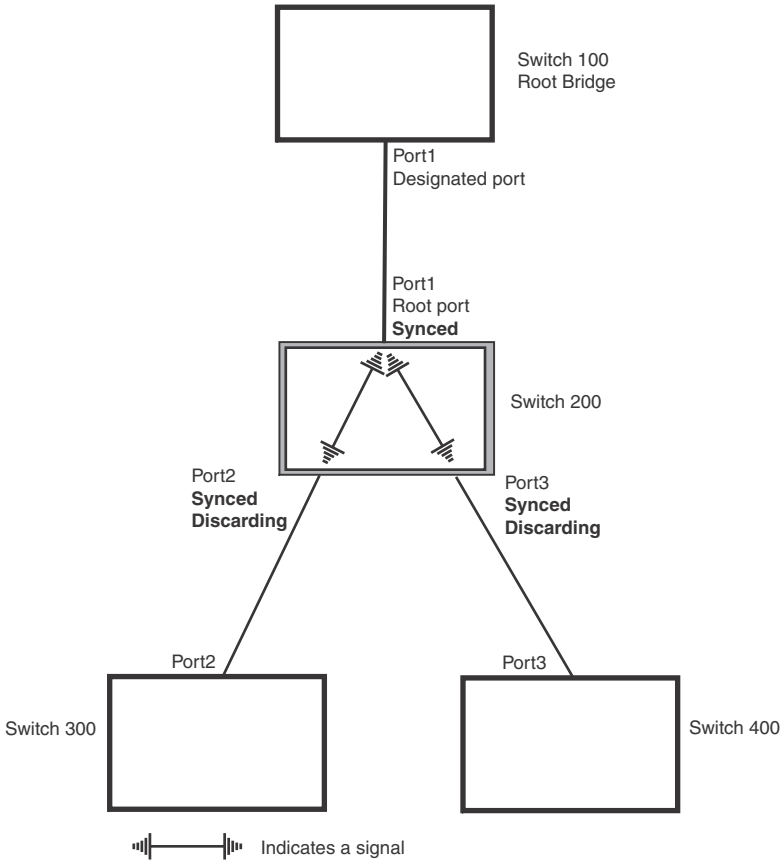
- **Sync** – Once the Root port is elected, it sets a sync signal on all the ports on the bridge. The signal tells the ports to synchronize their roles and states (Figure 44). Ports that are non-edge ports with a role of Designated port change into a discarding state. These ports have to negotiate with their peer ports to establish their new roles and states.

FIGURE 44 Sync stage



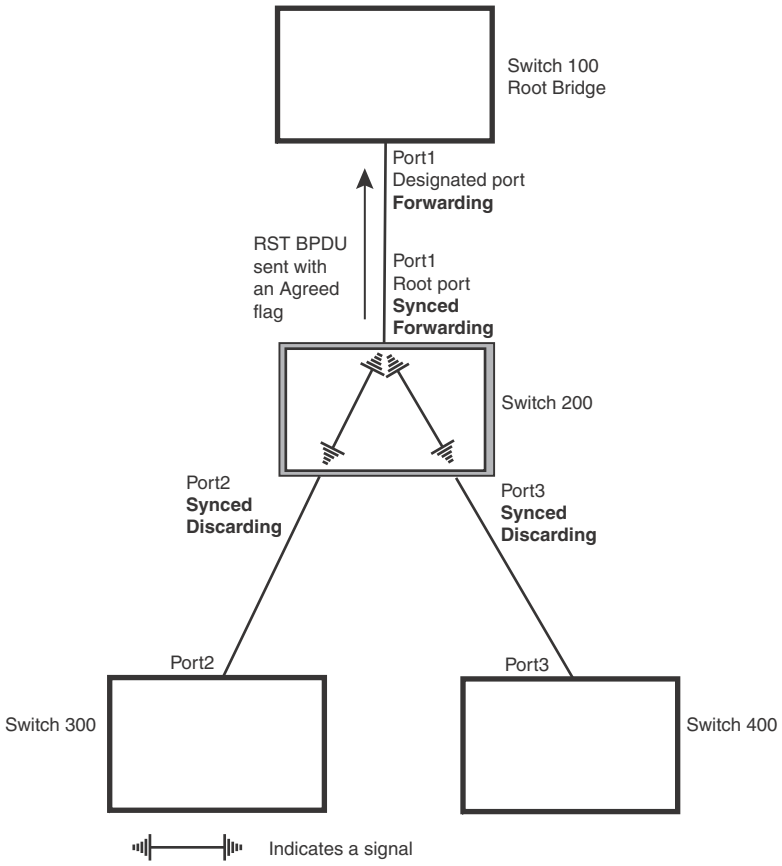
- **Synced** – Once the Designated port changes into a discarding state, it asserts a synced signal. Immediately, Alternate ports and Backup ports are synced. The Root port monitors the synced signals from all the bridge ports. Once all bridge ports asserts a synced signal, the Root port asserts its own synced signal (Figure 45).

FIGURE 45 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 46 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.

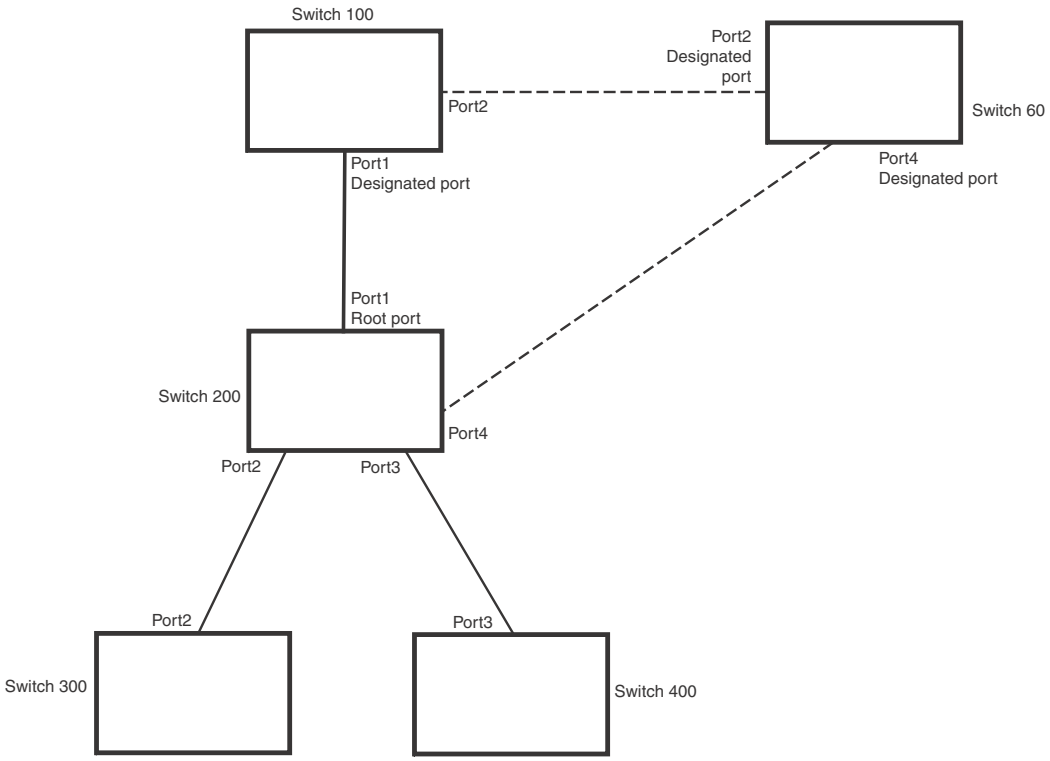
For example, Port2/Switch 200 sends an RST BPDU to Port2/Switch 300 that contains a proposal flag. Port2/Switch 300 asserts a proposed signal. Ports in Switch 300 then set sync signals on the ports to synchronize and negotiate their roles and states. Then the ports assert a synced signal and when the Root port in Switch 300 asserts it is 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 47, a new root bridge is added to the topology.

FIGURE 47 Addition of a new root bridge

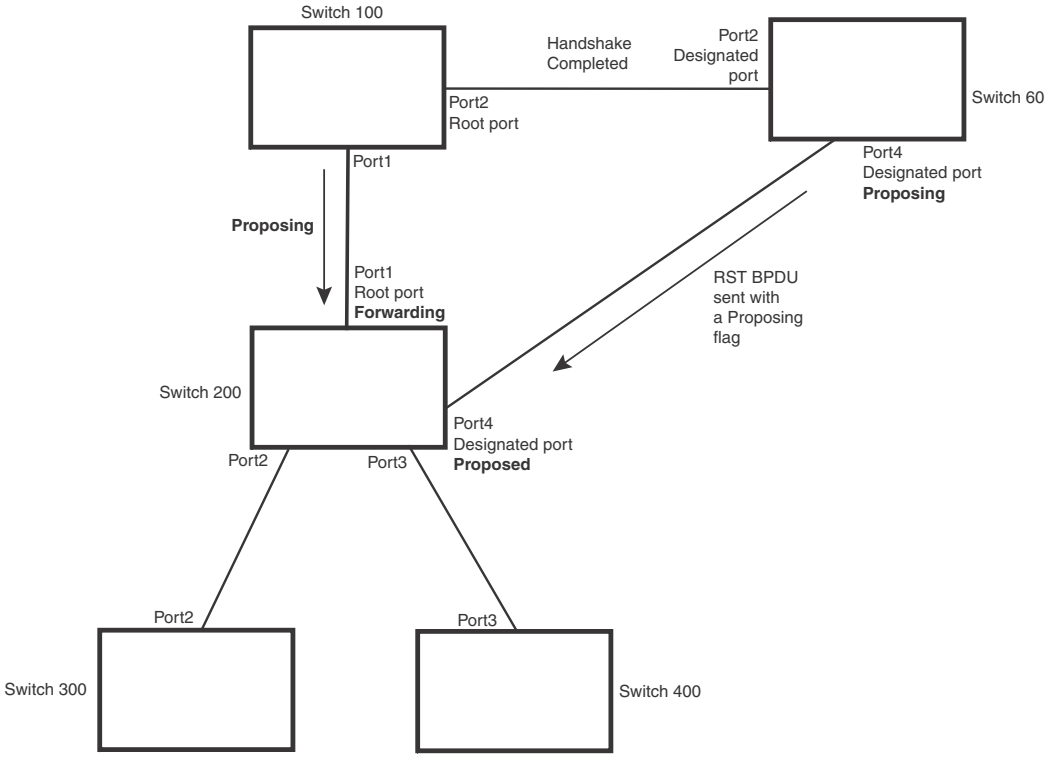


The handshake that occurs between Switch 60 and Switch 100 follows the one described in the previous section ("Handshake when no root port is elected" on page 353). The former root bridge becomes a non-root bridge and establishes a Root port (Figure 48).

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:

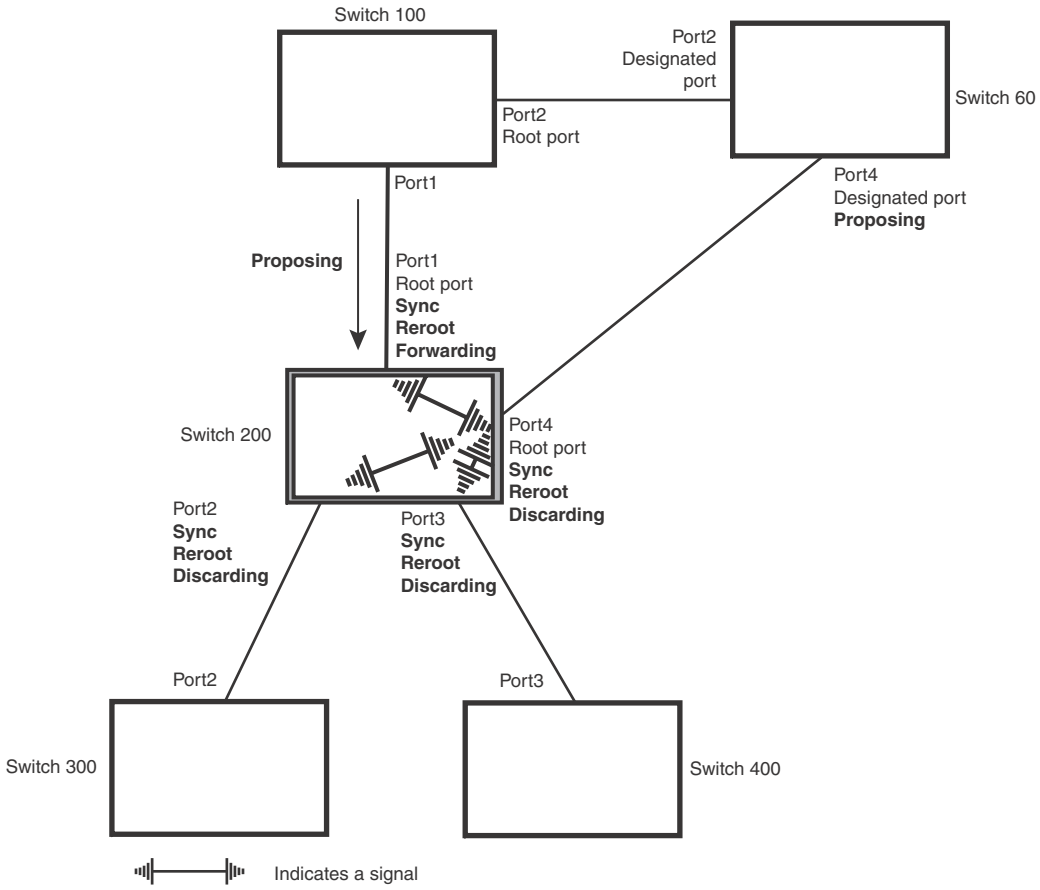
- **Proposing and Proposed** – The Designated port on the new root bridge (Port4/Switch 60) sends an RST BPDU that contains a proposing signal to Port4/Switch 200 to inform the port that it is ready to put itself in a forwarding state (Figure 48). RSTP algorithm determines that the RST BPDU that Port4/Switch 200 received is superior to what it can generate, so Port4/Switch 200 assumes a Root port role.

FIGURE 48 New root bridge sending a proposal flag



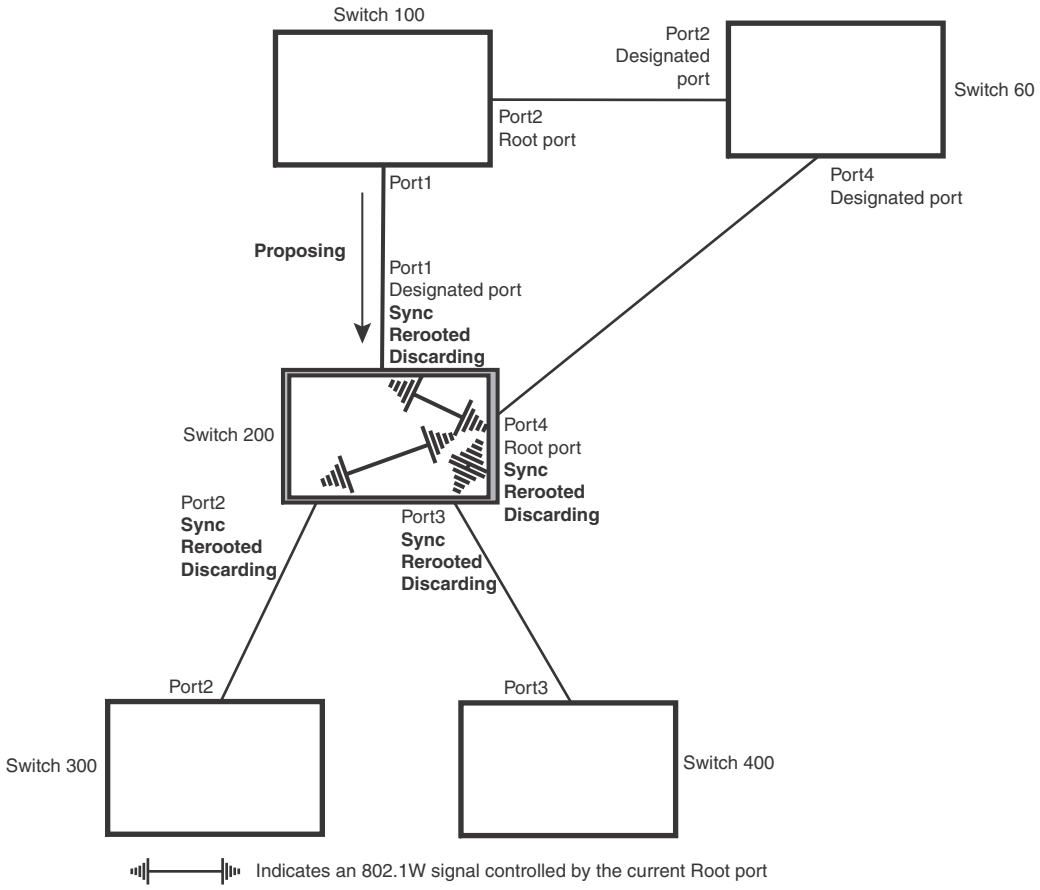
- **Sync and Reroot** – The Root port then asserts a sync and a reroot signal on all the ports on the bridge. The signal tells the ports that a new Root port has been assigned and they are to renegotiate their new roles and states. The other ports on the bridge assert their sync and reroot signals. Information about the old Root port is discarded from all ports. Designated ports change into discarding states (Figure 49).

FIGURE 49 Sync and reroot



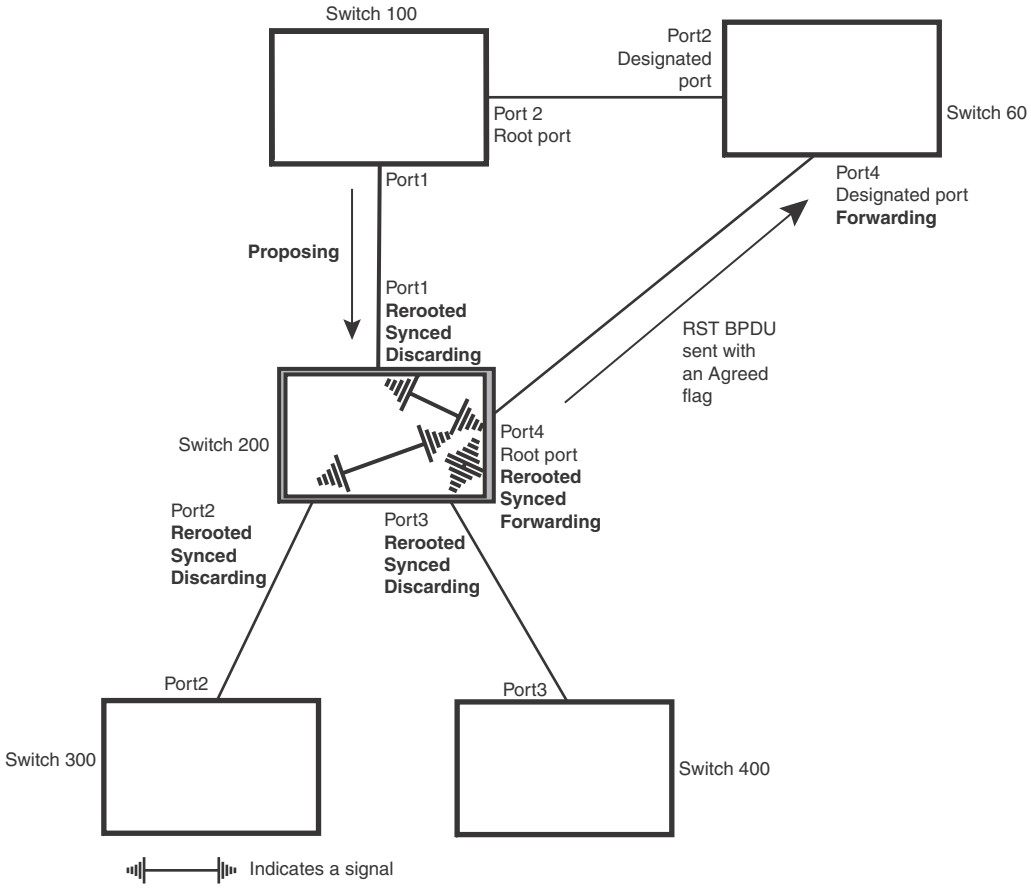
- **Sync and Rerooted** – When the ports on Switch 200 have completed the reroot phase, they assert their rerooted signals and continue to assert their sync signals as they continue in their discarding states. They also continue to negotiate their roles and states with their peer ports (Figure 50).

FIGURE 50 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 50). The Root port also moves into a forwarding state.

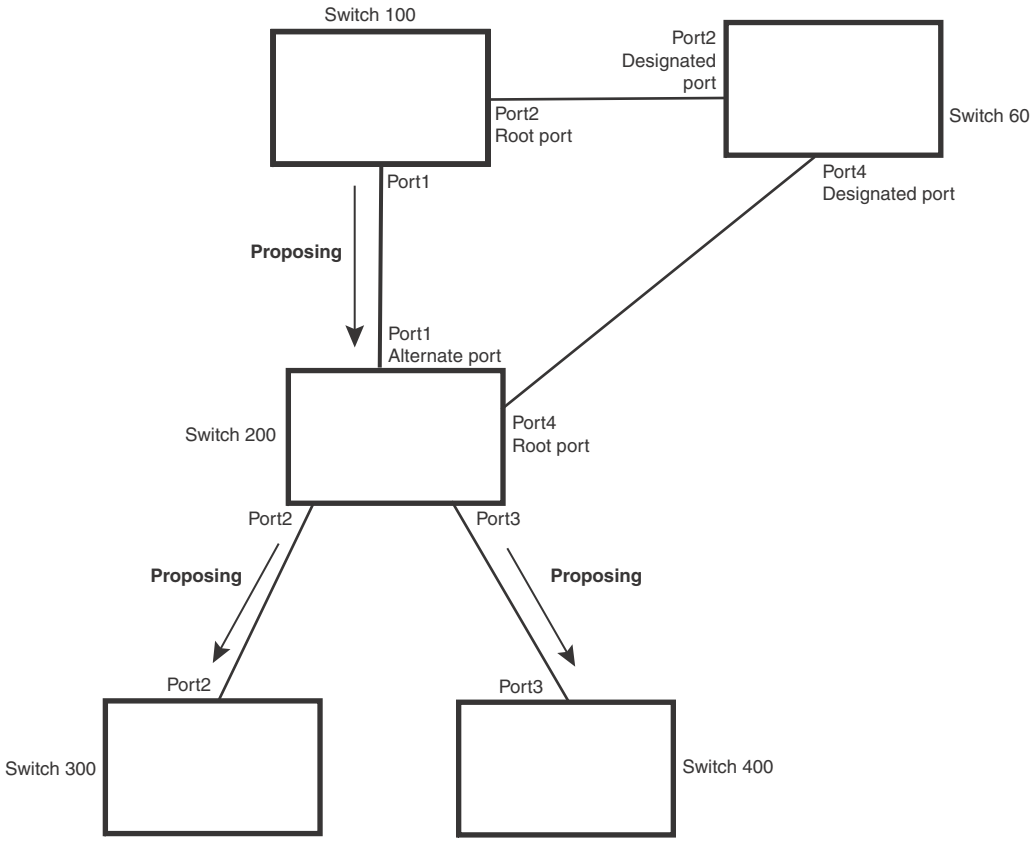
FIGURE 51 Rerooted,synced, and agreed



The old Root port on Switch 200 becomes an Alternate Port (Figure 52). 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 52 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.

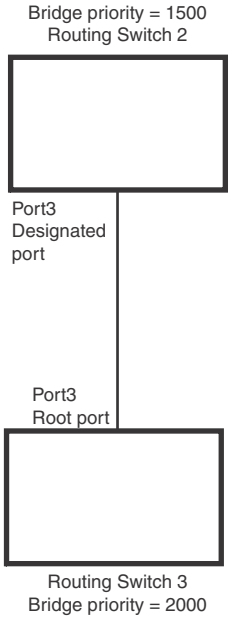
NOTE
The remaining examples assume that the appropriate handshake mechanisms occur as port roles and states change.

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.

Convergence at start up

In Figure 53, 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 53 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.

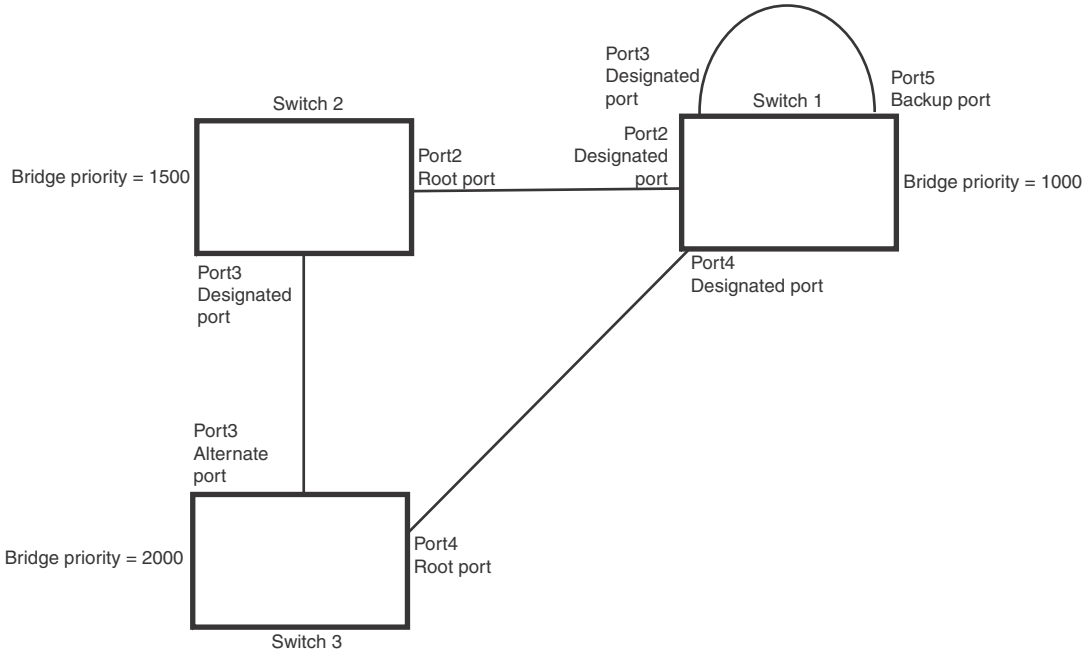
Port3/Switch 3, which starts with a role of Designated port, receives the RST BPDU and finds that it is superior to what it can transmit; therefore, Port3/Switch 3 assumes a new port role, that of a Root port. Port3/Switch 3 transmits an RST BPDU with an agreed flag back to Switch 2 and immediately goes 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 54).

FIGURE 54 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.

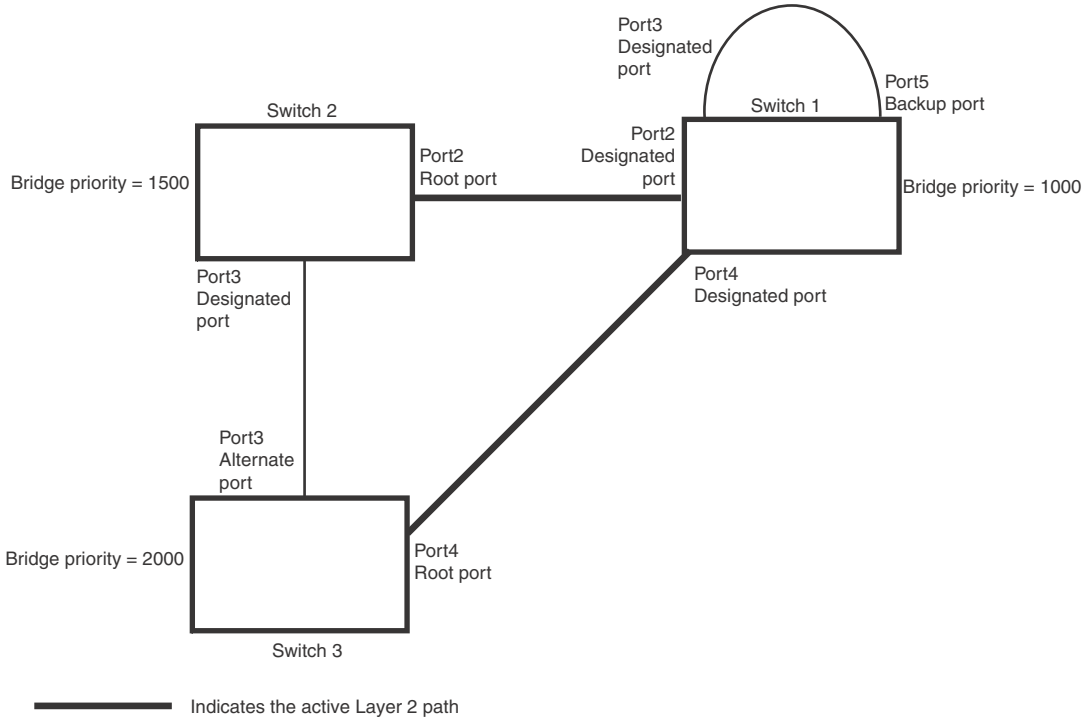
The Port2/Switch 2 bridge also sends an RST BPDU with an agreed flag Port2/Switch 1 that Port2 is the new Root port. Both ports go into forwarding states.

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 55.

FIGURE 55 Active Layer 2 path

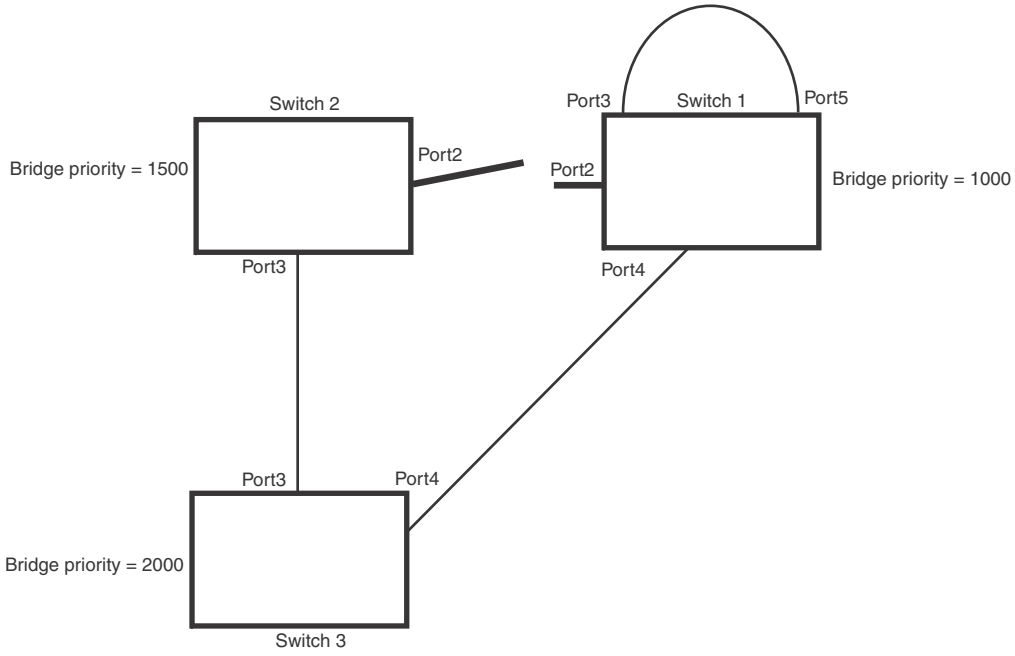


Convergence after a link failure

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 56).

FIGURE 56 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.

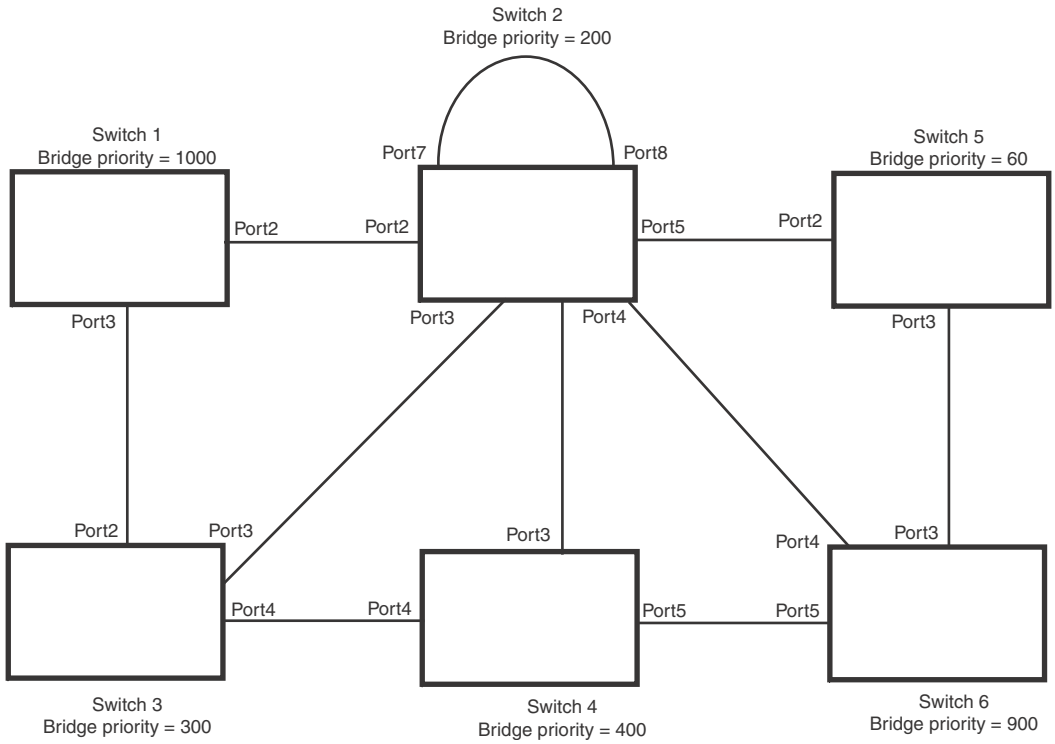
Now Port3/Switch 2 does not go into a forwarding state instantly like the Root port. It waits until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state. The wait, however, does not cause a denial of service, since the essential connectivity in the topology has already been established.

When fully restored, the topology is the same as that shown on Figure 54.

Convergence in a complex RSTP topology

The following is an example of a complex RSTP topology.

FIGURE 57 Complex RSTP topology



In Figure 57, 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.

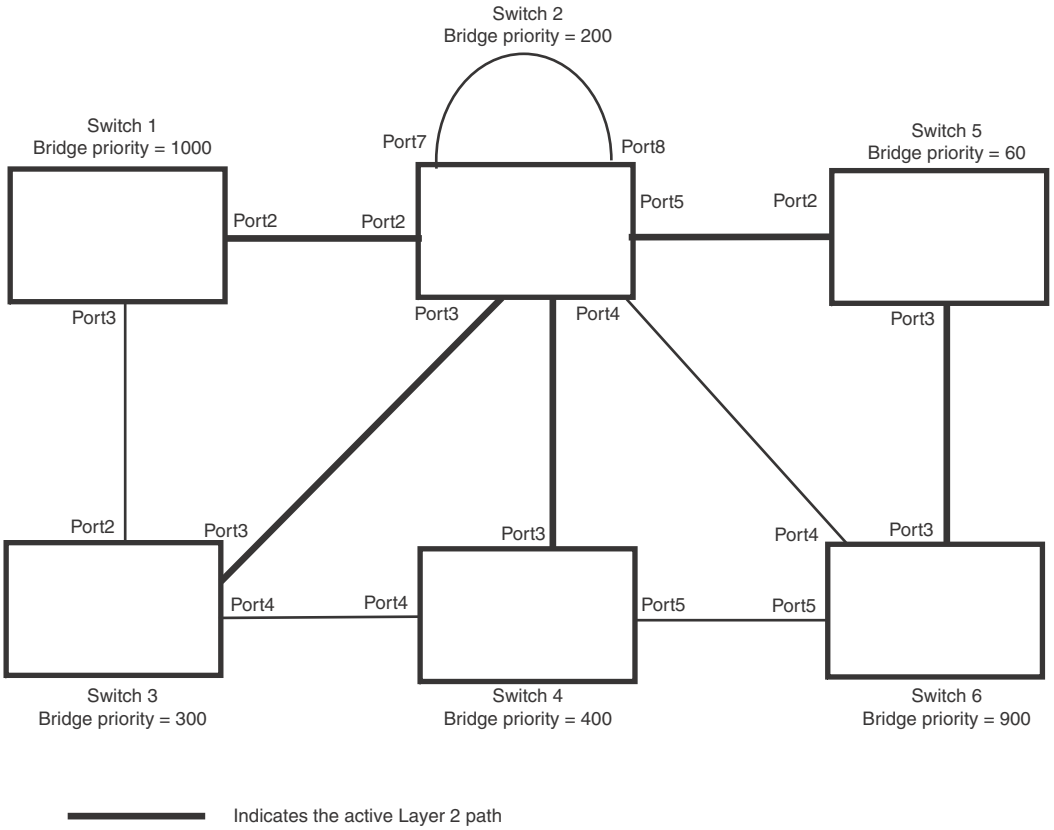
Port4/Switch 3 receives an RST BPDU that is inferior to what it can transmit; therefore, the port retains its Designated port role.

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 58 shows the active Layer 2 path of the topology in Figure 57.

FIGURE 58 Active Layer 2 path in complex topology



Propagation of topology change

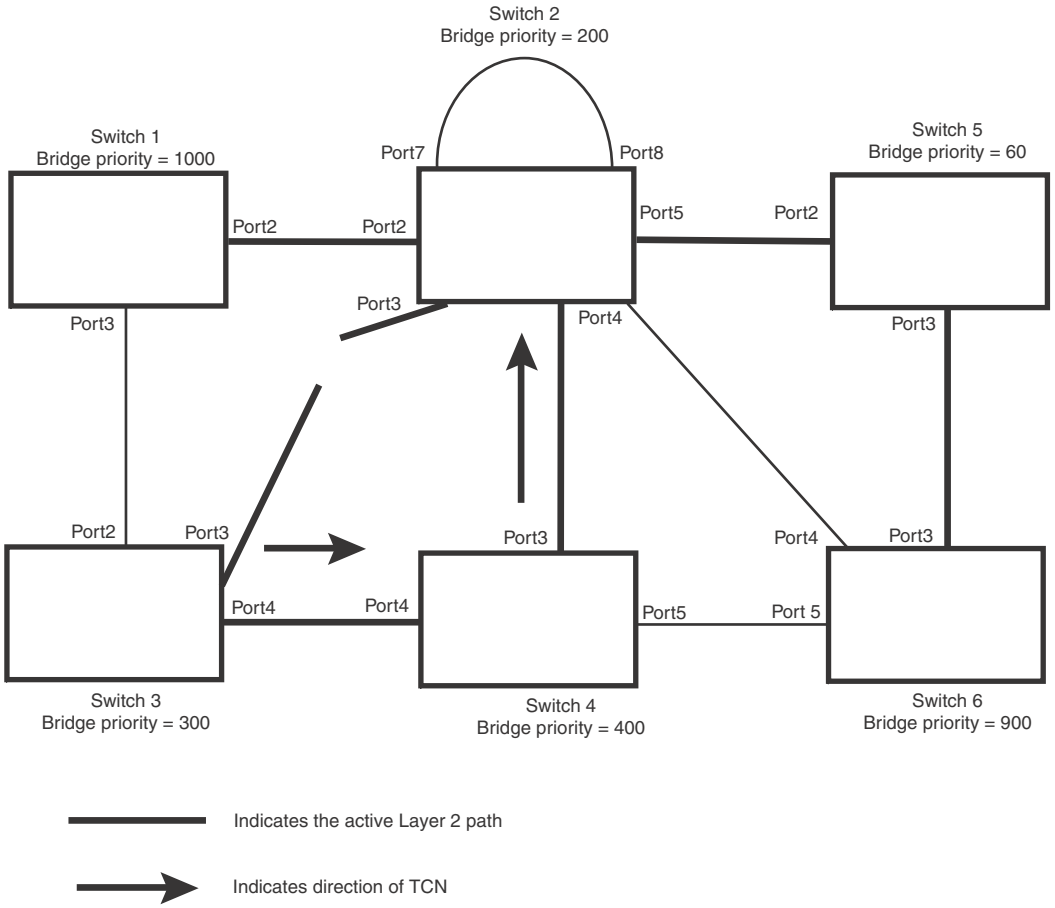
The Topology Change state machine generates and propagates the topology change notification messages on each port. When a Root port or a Designated port goes into a forwarding state, the Topology Change state machine on those ports send a topology change notice (TCN) to all the bridges in the topology 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 59, 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 59.)

FIGURE 59 Beginning of topology change notice

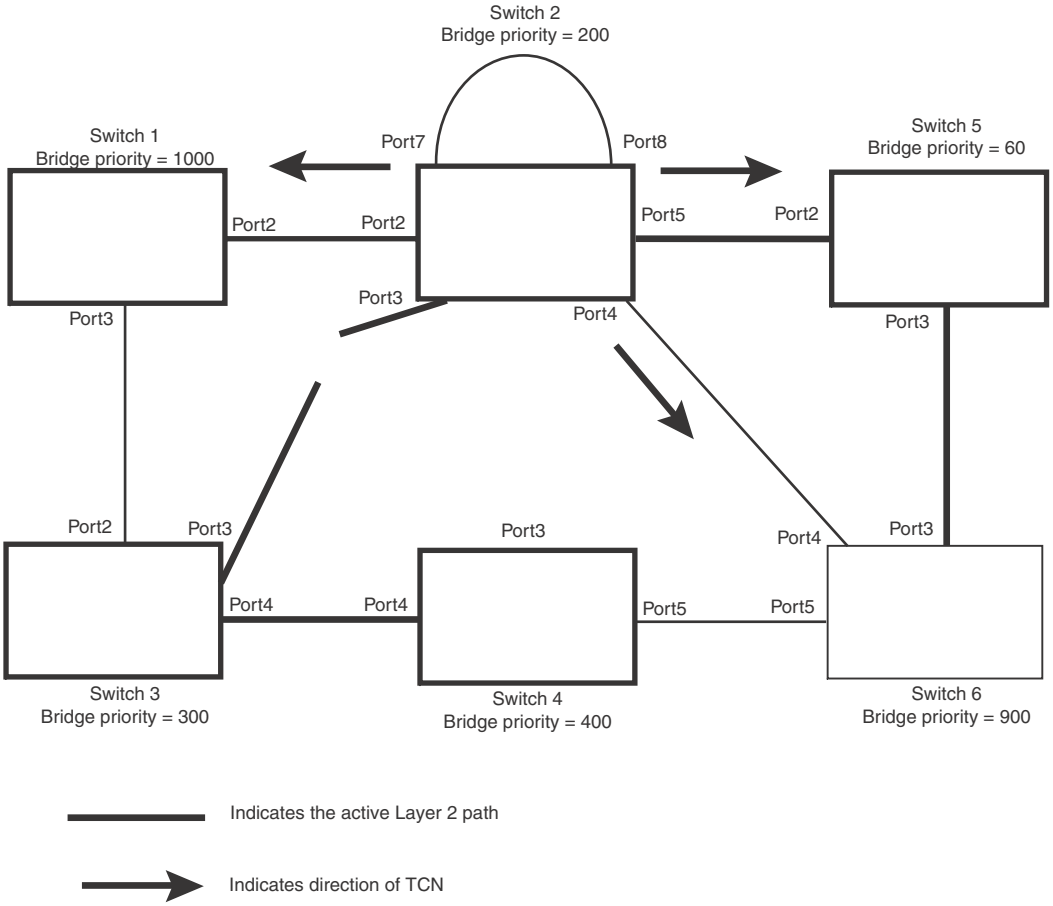


Switch 2 then starts the TCN timer on the Designated ports and sends RST BPDUs that contain the TCN as follows (Figure 60):

- Port5/Switch 2 sends the TCN to Port2/Switch 5
- Port4/Switch 2 sends the TCN to Port4/Switch 6

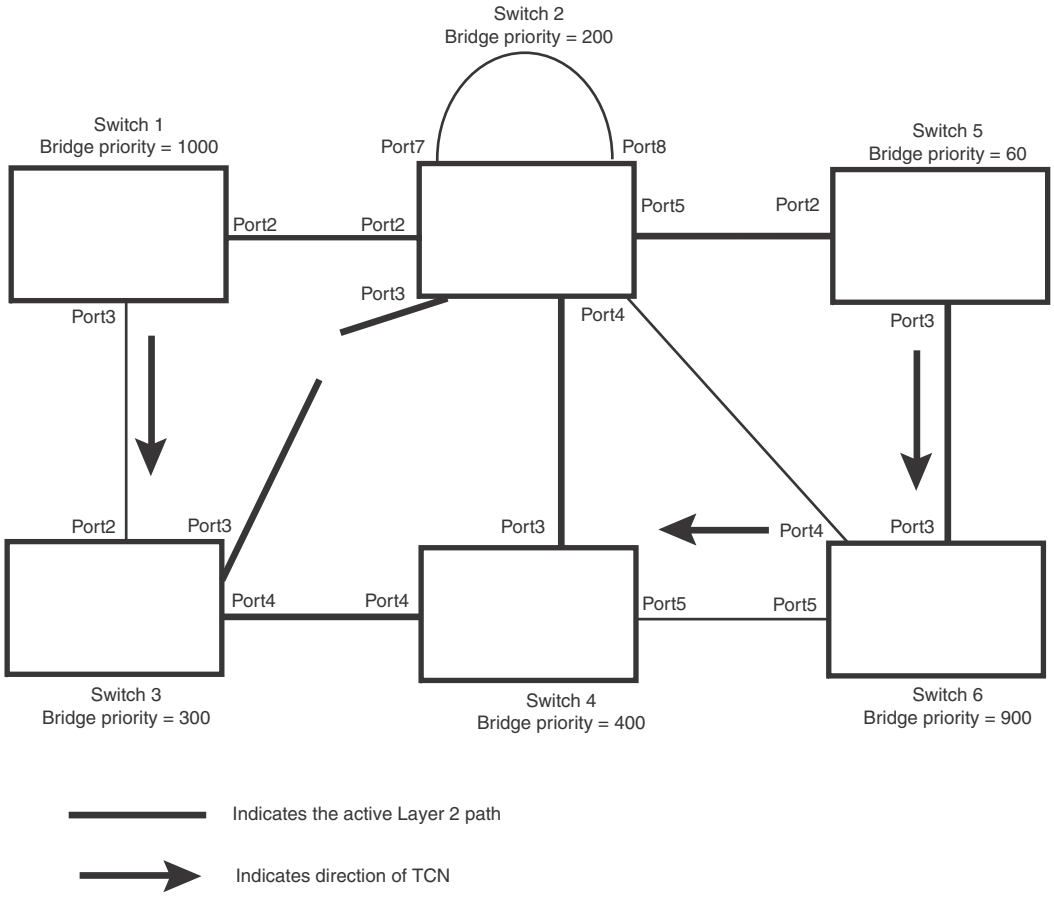
- Port2/Switch 2 sends the TCN to Port2/Switch 1

FIGURE 60 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 61).

FIGURE 61 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.

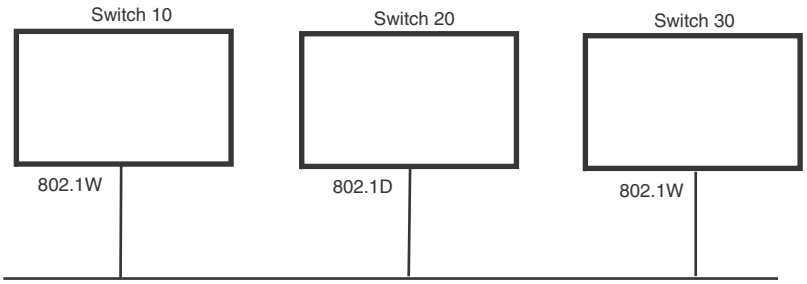
Compatibility with 802.1D means that an RSTP-enabled port can send BPDUs in the STP or 802.1D format when one of the following events occur:

- 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 62, 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 62 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 device.

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 trunk group, the change affects all ports in the trunk group.

Enabling or disabling RSTP in a port-based VLAN

Use the following procedure to disable or enable RSTP on a device 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.

```
BigIron RX(config)# vlan 10
BigIron RX(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.

```
BigIron RX(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 trunk group, the change affects all ports in that trunk group.

To disable or enable RSTP on a port, enter commands such as the following.

```
BigIron RX(config)# interface 1/1
BigIron RX(config-if-e1000-1/1)# no spanning-tree
```

Syntax: [no] spanning-tree [protect]

The value of **protect** will drop the BPDUs received on that specific interface.

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.

```
BigIron RX(config)# vlan 20
BigIron RX(config-vlan-20)# rstp priority 0
```

To make this change in the default VLAN, enter the following commands.

```
BigIron RX(config)# vlan 1
BigIron RX(config-vlan-1)# rstp priority 0
```

Syntax: spanning-tree 802-1w [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.

```
BigIron RX(config)# vlan 10
BigIron RX(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 76 shows the recommended path cost values from the IEEE standards.

TABLE 76 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

TABLE 76 Recommended path cost values of RSTP (Continued)

Link speed	Recommended (default) RSTP path cost values	Recommended RSTP path cost range
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 – 655352, in increments of 4. If you enter a value that is not divisible by four the software rounds to the nearest value that is. The default is 128 (the terminal shows this too). 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.

Example

Suppose you want to enable RSTP on a system with no active port-based VLANs and change the hello-time from the default value of 2 to 8 seconds. Additionally, suppose you want to change the path and priority costs for port 5 only. To do so, enter the following commands.

```
BigIron RX(config)# spanning-tree 802-1w hello-time 8
BigIron RX(config)# spanning-tree 802-1w ethernet 5 path-cost 15 priority 64
```

Fast port span

When STP is running on a device, message forwarding is delayed during the spanning tree recalculation period following a topology change. The STP forward delay parameter specifies the period of time a bridge waits before forwarding data packets. The forward delay controls the listening and learning periods of STP reconvergence. You can configure the forward delay to a value from 4 – 30 seconds. The default is 15 seconds. Thus, using the standard forward delay, convergence requires 30 seconds (15 seconds for listening and an additional 15 seconds for learning) when the default value is used.

This slow convergence is undesirable and unnecessary in some circumstances. The Fast Port Span feature allows certain ports to enter the forwarding state in four seconds. Specifically, Fast Port Span allows faster convergence on ports that are attached to end stations and thus do not present the potential to cause Layer 2 forwarding loops. Because the end stations cannot cause

forwarding loops, they can safely go through the STP state changes (blocking to listening to learning to forwarding) more quickly than is allowed by the standard STP convergence time. Fast Port Span performs the convergence on these ports in four seconds (two seconds for listening and two seconds for learning).

In addition, Fast Port Span enhances overall network performance in the following ways:

- Fast Port Span reduces the number of STP topology change notifications on the network. When an end station attached to a Fast Span port comes up or down, the Brocade device does not generate a topology change notification for the port. In this situation, the notification is unnecessary since a change in the state of the host does not affect the network's topology.
- Fast Port Span eliminates unnecessary MAC cache aging that can be caused by topology change notifications. Bridging devices age out the learned MAC addresses in their MAC caches if the addresses are unrefreshed for a given period of time, sometimes called the MAC aging interval. When STP sends a topology change notification, devices that receive the notification use the value of the STP forward delay to quickly age out their MAC caches. For example, if a device's normal MAC aging interval is 5 minutes, the aging interval changes temporarily to the value of the forward delay (for example, 15 seconds) in response to an STP topology change.

  In normal STP, the accelerated cache aging occurs even when a single host goes up or down. Because Fast Port Span does not send a topology change notification when a host on a Fast Port Span port goes up or down, the unnecessary cache aging that can occur in these circumstances under normal STP is eliminated.

Fast Port Span is a system-wide parameter and is enabled by default. Thus, when you boot a device, all the ports that are attached only to end stations run Fast Port Span. For ports that are not eligible for Fast Port Span, such as ports connected to other networking devices, the device automatically uses the normal STP settings. If a port matches any of the following criteria, the port is ineligible for Fast Port Span and uses normal STP instead:

- The port is 802.1q tagged
- The port is a member of a trunk group
- The port has learned more than one active MAC address
- An STP Configuration BPDU has been received on the port, thus indicating the presence of another bridge on the port.

You also can explicitly exclude individual ports from Fast Port Span if needed. For example, if the only uplink ports for a wiring closet switch are Gigabit ports, you can exclude the ports from Fast Port Span.

Disabling and re-enabling fast port span

Fast Port Span is a system-wide parameter and is enabled by default. Thus all ports that are eligible for Fast Port Span use it.

To disable or re-enable Fast Port Span, use the following method.

Using the CLI

To disable Fast Port Span, enter the following commands.

```
BigIron RX(config)# no fast port-span
BigIron RX(config)# write memory
```

Syntax: [no] fast port-span

NOTE
The **fast port-span** command has additional parameters that let you exclude specific ports. These parameters are shown in the following section.

To re-enable Fast Port Span, enter the following commands.

```
BigIron RX(config)# fast port-span
BigIron RX(config)# write memory
```

Excluding specific ports from fast port span

You can exclude individual ports from Fast Port Span while leaving Fast Port Span enabled globally. To do so, use the following method.

Using the CLI

To exclude a port from Fast Port Span, enter commands such as the following.

```
BigIron RX(config)# fast port-span exclude ethernet 1/1
BigIron RX(config)# write memory
```

To exclude a set of ports from Fast Port Span, enter commands such as the following.

```
BigIron RX(config)# fast port-span exclude ethernet 1/1 ethernet 2/1 ethernet 3/2
BigIron RX(config)# write memory
```

To exclude a contiguous (unbroken) range of ports from Fast Span, enter commands such as the following.

```
BigIron RX(config)# fast port-span exclude ethernet 1/1 to 1/24
BigIron RX(config)# write memory
```

Syntax: [no] fast port-span [exclude ethernet <*portnum*> [ethernet <*portnum*>... | to <*portnum*>]]

To re-enable Fast Port Span on a port, enter a command such as the following.

```
BigIron RX(config)# no fast port-span exclude ethernet 1/1
BigIron RX(config)# write memory
```

This command re-enables Fast Port Span on port 1/1 only and does not re-enable Fast Port Span on other excluded ports. You also can re-enable Fast Port Span on a list or range of ports using the syntax shown above this example.

To re-enable Fast Port Span on all excluded ports, disable and then re-enable Fast Port Span by entering the following commands.

```
BigIron RX(config)# no fast port-span
BigIron RX(config)# fast port-span
BigIron RX(config)# write memory
```

Disabling and then re-enabling Fast Port Span clears the exclude settings and thus enables Fast Port Span on all eligible ports. To make sure Fast Port Span remains enabled on the ports following a system reset, save the configuration changes to the startup-config file after you re-enable Fast Port Span. Otherwise, when the system resets, those ports will again be excluded from Fast Port Span.

Fast uplink span

The Fast Port Span feature described in the previous section enhances STP performance for end stations. The Fast Uplink feature enhances STP performance for wiring closet switches with redundant uplinks. Using the default value for the standard STP forward delay, convergence following a transition from an active link to a redundant link can take 30 seconds (15 seconds for listening and an additional 15 seconds for learning).

You can use the Fast Uplink feature on a Brocade device deployed as a wiring closet switch to decrease the convergence time for the uplink ports to another device to just four seconds (two seconds for listening and two seconds for learning). The wiring closet switch must be a Brocade device but the device at the other end of the link can be a Brocade device or another vendor's switch. Configuration of the Fast Uplink Span feature takes place entirely on the Brocade device.

To configure the Fast Uplink Span feature, specify a group of ports that have redundant uplinks on the wiring closet switch (Brocade device) as members of a Fast Uplink Group. If the active link becomes unavailable, the Fast Uplink Span feature transitions the forwarding to one of the other ports in four seconds. You can configure one Fast Uplink Span group on the device. All Fast Uplink Span ports are members of the same Fast Uplink Span group.

NOTE
To avoid the potential for temporary bridging loops, Brocade recommends that you use the Fast Uplink feature only for wiring closet switches (switches at the edge of the network cloud). In addition, enable the feature only on a group of ports intended for redundancy, so that at any given time only one of the ports is expected to be in the forwarding state.

NOTE
When the BigIron RX first comes up or when STP is first enabled, the uplink ports still must go through the standard STP state transition without any acceleration. This behavior guards against temporary routing loops as the switch tries to determine the states for all the ports. Fast Uplink Span acceleration applies only when a working uplink becomes unavailable.

Fast uplink span rules for trunk groups

If you add a port to a Fast Uplink Span group that is a member of a trunk group, the following rules apply:

- If you add the primary port of a trunk group to the Fast Uplink Span group, all other ports in the trunk group are automatically included in the group. Similarly, if you remove the primary port in a trunk group from the Fast Uplink Span group, the other ports in the trunk group are automatically removed from the Fast Uplink Span group.
- You cannot add a subset of the ports in a trunk group to the Fast Uplink Span group. All ports in a trunk group have the same Fast Uplink Span property, as they do for other port properties.
- If the working trunk group is partially down but not completely down, no switch-over to the backup occurs. This behavior is the same as in the standard STP feature.
- If the working trunk group is completely down, a backup trunk group can go through an accelerated transition only if the following are true:
  - The trunk group is included in the fast uplink group.
  - All other ports except those in this trunk group are either disabled or blocked. The accelerated transition applies to all ports in this trunk group.

- When the original working trunk group comes back (partially or fully), the transition back to the original topology is accelerated if the conditions listed above are met.

Configuring a fast uplink port group

To enable Fast Uplink, use the following method.

Using the CLI

To configure a group of ports for Fast Uplink Span, enter the following commands.

```
BigIron RX(config)# fast uplink-span ethernet 4/1 to 4/4
BigIron RX(config)# write memory
```

Syntax: [no] fast uplink-span [ethernet <*portnum*> [ethernet <*portnum*>... | to <*portnum*>]]

This example configures four ports, 4/1 – 4/4, as a Fast Uplink Span group. In this example, all four ports are connected to a wiring closet switch. Only one of the links is expected to be active at any time. The other links are redundant. For example, if the link on port 4/1 is the active link on the wiring closet switch but becomes unavailable, one of the other links takes over. Because the ports are configured in a Fast Uplink Span group, the STP convergence takes about four seconds instead of taking 30 seconds or longer using the standard STP forward delay.

If you add a port that is the primary port of a trunk group, all ports in the trunk group become members of the Fast Uplink Span group.

You can add ports to a Fast Uplink Span group by entering the **fast uplink-span** command additional times with additional ports. The device can have only one Fast Uplink Span group, so all the ports you identify as Fast Uplink Span ports are members of the same group.

To remove a Fast Uplink Span group or to remove individual ports from a group, use “no” in front of the appropriate **fast uplink-span** command. For example, to remove ports 4/3 and 4/4 from the Fast Uplink Span group configured above, enter the following commands.

```
BigIron RX(config)# no fast uplink-span ethernet 4/3 to 4/4
BigIron RX(config)# write memory
```

If you delete a port that is the primary port of a trunk group, all ports in the trunk group are removed from the Fast Uplink Span group.

Displaying RSTP information

You can display a summary or details of the RSTP information.

To display a summary of RSTP, use the following command.

```
BigIron RX(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 77.

TABLE 77 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.

TABLE 77 CLI display of RSTP summary (Continued)

This field...	Displays...
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.
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.

TABLE 77 CLI display of RSTP summary (Continued)

This field...	Displays...
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 347 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 351 and "Edge port and non-edge port states" on page 352.
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.

```
BigIron RX(config)#show rstp detail
VLAN 10 - RSTP instance 0
------------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:

BridgeId 0001000480a04000, RootBridgeId 0001000480a04000
Control ports - ethe 1/3 ethe 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.

TABLE 78 The show rstp detail command output

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

TABLE 78 The show rstp detail command output (Continued)

This field...	Displays...
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 347 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 351 and "Edge port and non-edge port states" on page 352.

Chapter

Metro Ring Protocol (MRP) Phase 1 and 2

14

In this chapter

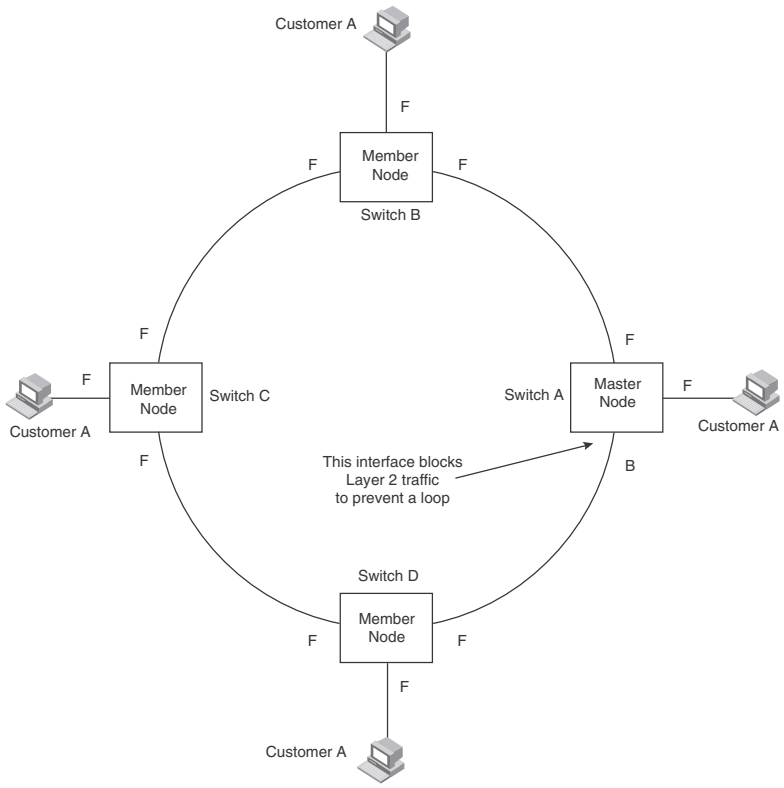
Metro Ring Protocol (MRP) phase 1

MRP Phase 1 is a Brocade proprietary protocol that prevents Layer 2 loops and provides fast reconvergence in Layer 2 ring topologies. It is an alternative to STP and is especially useful in Metropolitan Area Networks (MANs) where using STP has the following drawbacks:

- STP allows a maximum of seven nodes. Metro rings can easily contain more nodes than this.
- STP has a slow reconvergence time, taking many seconds or even minutes. MRP can detect and heal a break in the ring in sub-second time.

Figure 63 shows an MRP metro ring.

FIGURE 63 Metro ring – normal state



The ring in this example consists of four MRP nodes (Brocade switches). Each node has two interfaces with the ring. Each node also is connected to a separate customer network. The nodes forward Layer 2 traffic to and from the customer networks through the ring. The ring interfaces are all in one port-based VLAN. Each customer interface can be in the same VLAN as the ring or in a separate VLAN.

One node, is configured as the master node of the MRP ring. One of the two 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. The secondary interface blocks the packet to prevent a Layer 2 loop.

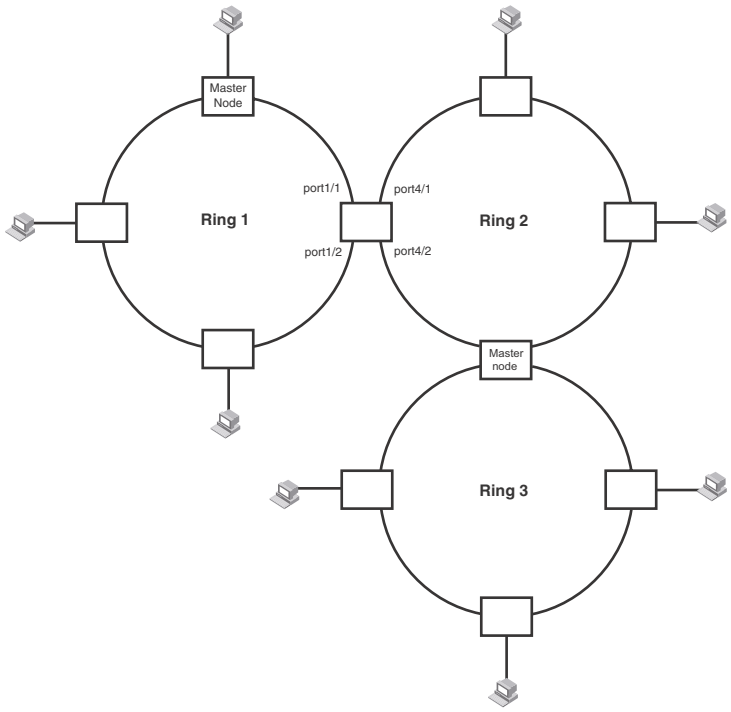
NOTE
When you configure MRP, Brocade recommends that you disable one of the ring interfaces before beginning the ring configuration. Disabling an interface prevents a Layer 2 loop from occurring while you are configuring MRP on the ring nodes. Once MRP is configured and enabled on all the nodes, you can re-enable the interface.

MRP rings without shared interfaces

MRP Phase 1 allows you to configure multiple MRP rings, as shown in Figure 64, but the rings cannot share the same link. For example, you cannot configure ring 1 and ring 2 to each have interfaces 1/1 and 1/2.

Also, when you configured an MRP ring, any node on the ring that can be designated as the master node for the ring. A master node can be the master node of more than one ring. (Refer to Figure 64.) Each ring is an independent ring and RHP packets are processed within each ring.

FIGURE 64 Metro ring – multiple rings

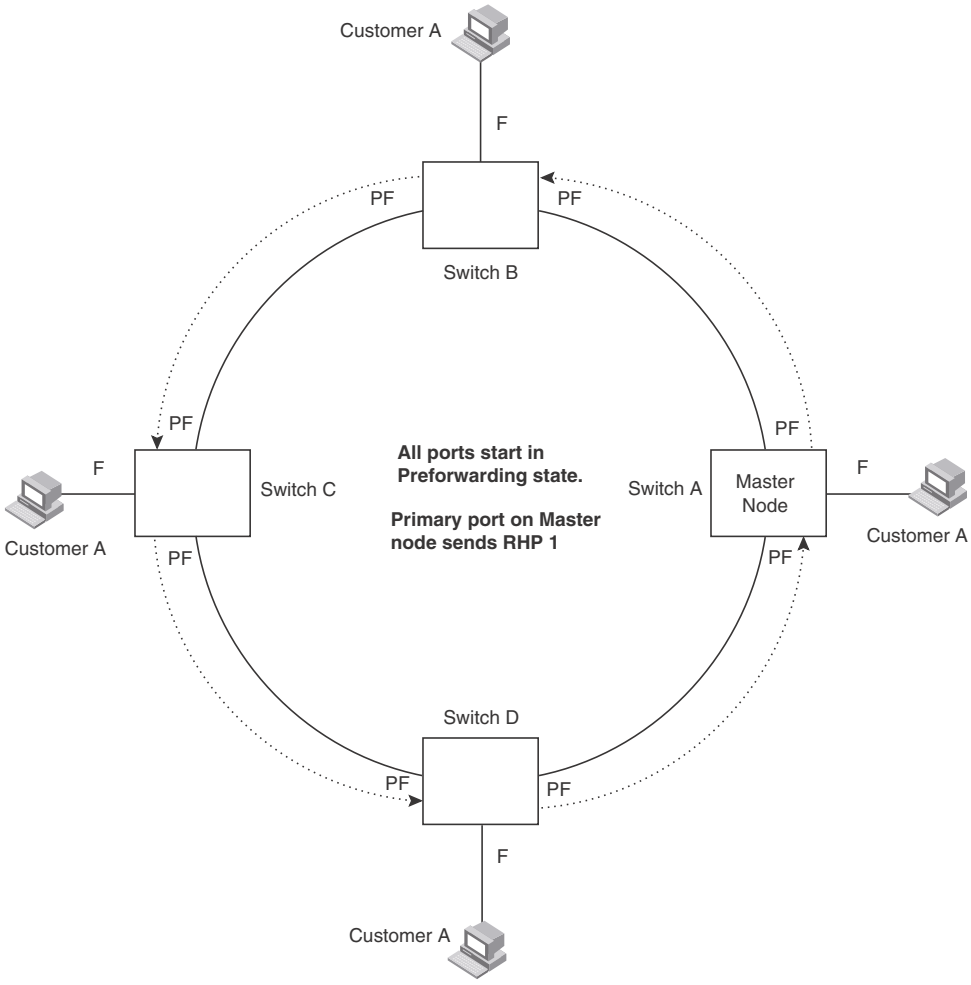


In this example, two nodes are each configured with two MRP rings. Any node in a ring can be the master for its ring. A node also can be the master for more than one ring.

Ring initialization

The ring shown in Figure 63 shows the port states in a fully initialized ring without any broken links. Figure 65 shows the initial state of the ring, when MRP is first enabled on the ring's switches. All ring interfaces on the master node and member nodes begin in the Preforwarding state (PF).

FIGURE 65 Metro ring – initial state



MRP uses Ring Health Packets (RHPs) to monitor the health of the ring. An RHP is an MRP protocol packet. 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 port depends on the RHPs.

A ring interface can have one of the following MRP states:

- **Preforwarding (PF)** – The interface can forward RHPs but cannot forward data. All ring ports being in this state when you enable MRP.

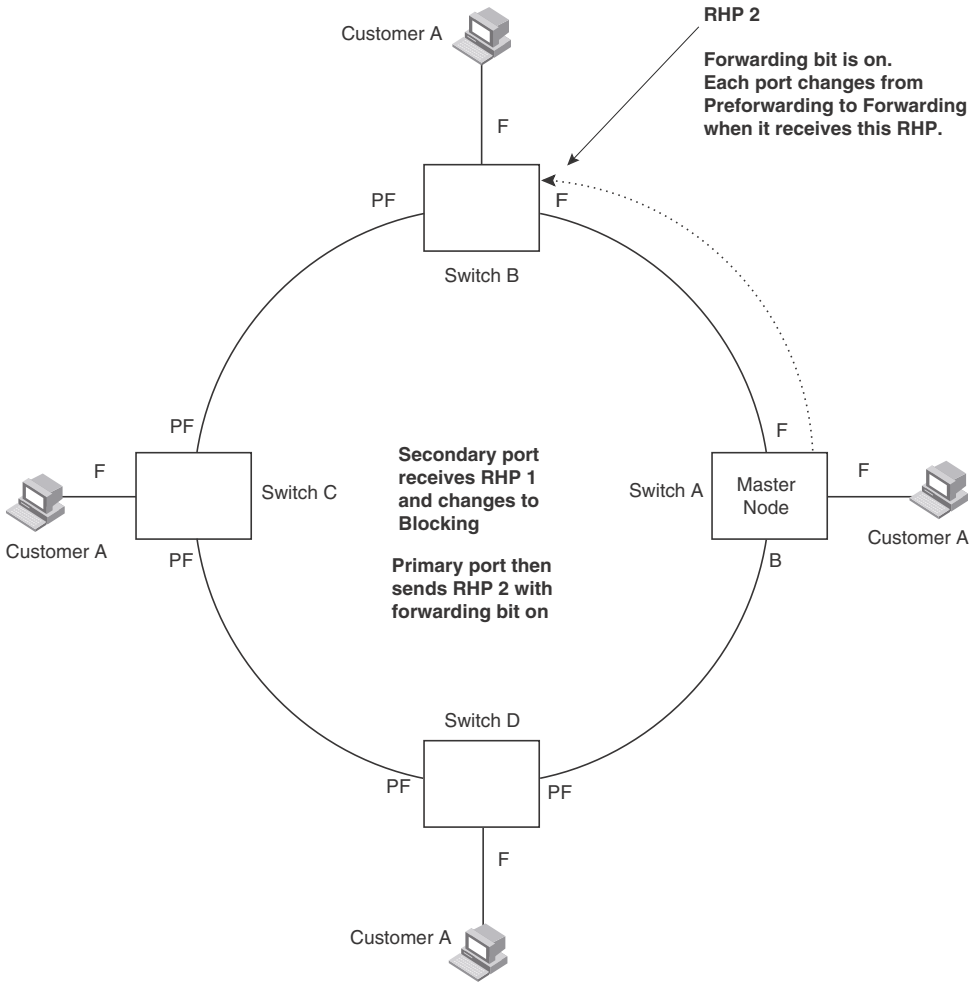
- **Forwarding (F)** – The interface can forward data as well as RHPs. An interface changes from Preforwarding to Forwarding when the port's preforwarding time expires. This occurs if the port does not receive an RHP from the Master, or if the forwarding bit in the RHPs received by the port is off. This indicates a break in the ring. The port heals the ring by changing its state to Forwarding. The preforwarding time is the number of milliseconds the port will remain in the Preforwarding state before changing to the Forwarding state, even without receiving an RHP.
- **Blocking (B)** – The interface can process RHPs, but cannot forward data. Only the secondary interface on the Master node can be Blocking.

When MRP is enabled, all ports begin in the Preforwarding state. The primary interface on the Master node, although it is in the Preforwarding state like the other ports, immediately sends an RHP onto the ring. The secondary port on the Master node listens for the RHP.

- If the secondary port receives the RHP, all links in the ring are up and the port changes its state to Blocking. The primary port then sends another MRP with its forwarding bit set on. As each of the member ports receives the RHP, the ports changes their state to Forwarding. Typically, this occurs in sub-second time. The ring very quickly enters the fully initialized state.
- If the secondary port does not receive the RHP by the time the preforwarding time expires, a break has occurred in the ring. The port changes its state to Forwarding. The member ports also change their states from Preforwarding to Forwarding as their preforwarding timers expire. The ring is not intact, but data can still travel among the nodes using the links that are up.

Figure 66 shows an example.

FIGURE 66 Metro ring – from Preforwarding to Forwarding

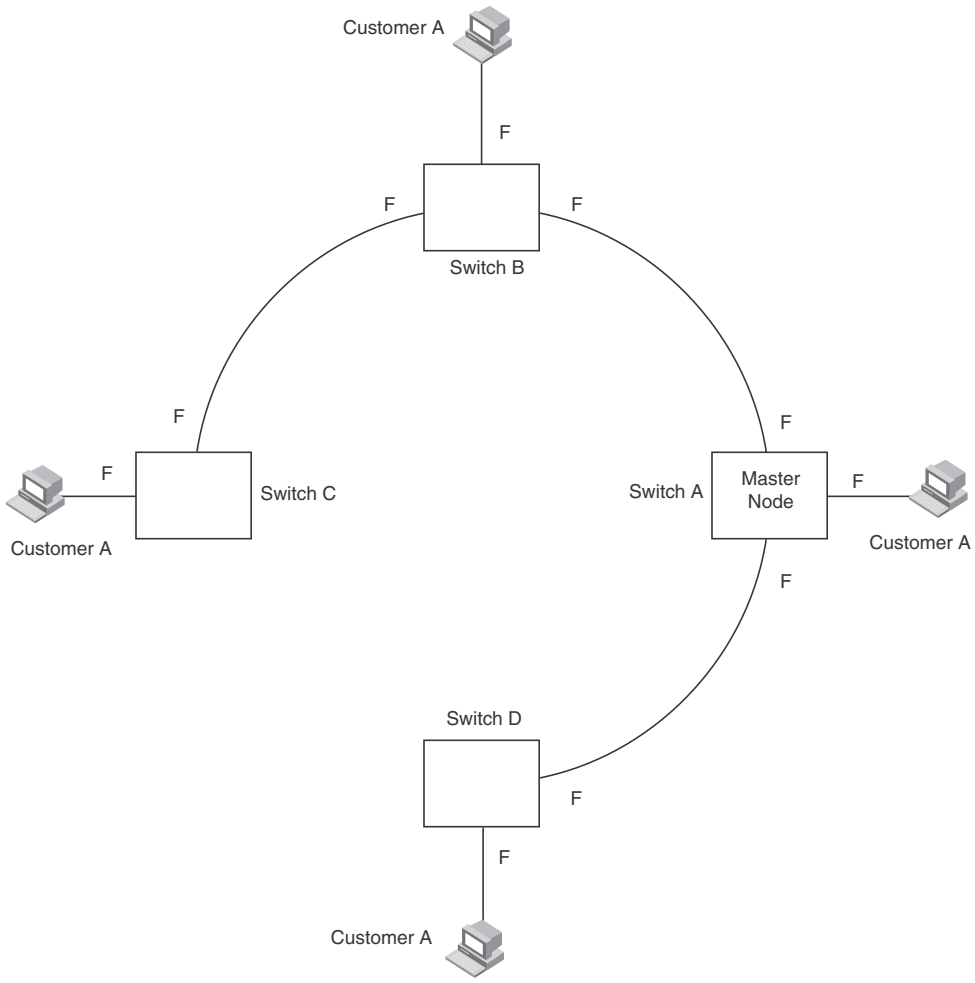


Each RHP also has a sequence number. MRP can use the sequence number to determine the round-trip time for RHPs in the ring. Refer to "MRP phase 2" on page 397.

How ring breaks are detected and healed

Figure 67 Shows the ring forwarding state following a link break. MRP quickly heals the ring and preserves connectivity among the customer networks.

FIGURE 67 Metro 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** – The Blocking interface on the Master node has a dead timer. If the dead time expires before the interface receives one of its ring's RHPs, the interface changes state to Preforwarding. Once the secondary interface changes state to Preforwarding:
  - If the interface receives an RHP, the interface changes back to the Blocking state and resets the dead 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 67.
- **Forwarding interfaces** – Each member interface remains in the Forwarding state.

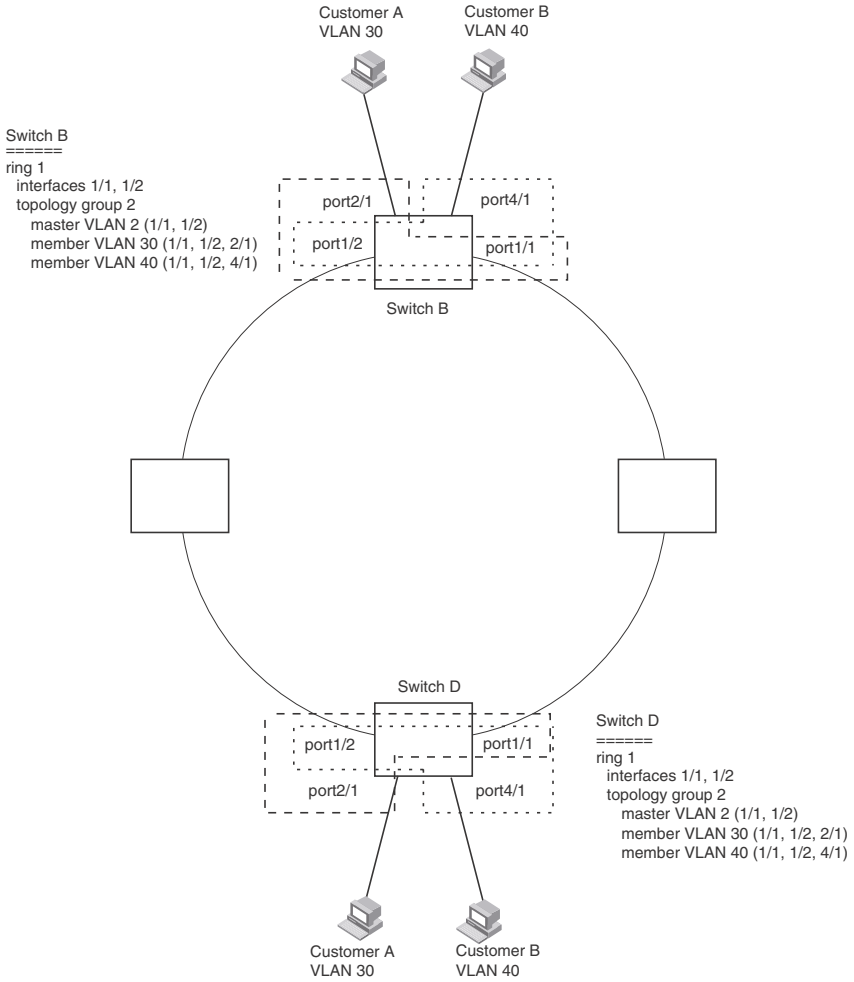
When the broken link is repaired, the link's interfaces come up in the Preforwarding state, which allows RHPs to travel through the restored interfaces and reach the secondary interface on the Master node.

- 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.

Master VLANs and customer VLANs in a topology group

All the ring ports must be in the same VLAN. Placing the ring ports in the same VLAN provides Layer 2 connectivity for a given customer across the ring. Figure 68 shows an example.

FIGURE 68 Metro ring – ring VLAN and customer VLANs



Notice that each customer has their own VLAN. Customer A has VLAN 30 and Customer B has VLAN 40. Customer A's host attached to Switch D can reach the Customer A host attached to Switch B at Layer 2 through the ring. Since Customer A and Customer B are on different VLANs, they will not receive each other's traffic.

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 68, 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 ports must be tagged, since they will be shared by multiple VLANs.
- The member VLAN for a customer must contain the two ring interfaces and the interfaces for the customer. Since these interfaces are shared with the master VLAN, they must be tagged. Do not add another customer's interfaces to the VLAN.

For more information about topology groups, refer to Chapter 16, "Topology Groups".

Refer to "MRP CLI example" on page 407 for the configuration commands required to implement the MRP configuration shown in Figure 68.

Configuring MRP

To configure MRP, perform the following tasks. You need to perform the first task on only one of the nodes. Perform the remaining tasks on all the nodes:

- Disable one of the ring interfaces. This prevents a Layer 2 loop from occurring while you are configuring the devices for MRP.
- 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:
  - Optionally, specify a name for the ring.
  - On the master node only, enable the device to be the master for the ring. Each ring can have only one master node.
  - Specify the MRP interfaces. Each device has two interfaces to an MRP ring.
  - Optionally, change the hello time and the preforwarding time. These parameters control how quickly failover occurs following a change in the state of a link in the ring.
  - Enable the ring.
- Optionally, add the ring's VLAN to a topology group to add more VLANs to the ring. If you use a topology group, make sure you configure MRP on the group's master VLAN. Refer to Chapter 16, "Topology Groups".
- Re-enable the interface you disabled to prevent a Layer 2 loop. Once MRP is enabled, MRP will prevent the Layer 2 loop.

NOTE
When MRP and UDLD are running together, Brocade recommends keeping the MRP preforwarding interval slightly higher than default(300ms) to 400 or 500ms to prevent the possibility of a temporary loop of a few milliseconds.

Adding an MRP ring to a VLAN

NOTE
If you plan to use a topology group to add VLANs to the ring, make sure you configure MRP on the topology group's master VLAN.

To add an MRP ring to a VLAN, enter commands such as the following.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# metro-ring 1
BigIron RX(config-vlan-2-mrp-1)# name CustomerA
BigIron RX(config-vlan-2-mrp-1)# master
BigIron RX(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
BigIron RX(config-vlan-2-mrp-1)# enable
```

These commands configure an MRP ring on VLAN 2. The ring ID is 1, the ring name is CustomerA, and this node (this device) is the master for the ring. The ring interfaces are 1/1 and 1/2. Interface 1/1 is the primary interface and 1/2 is the secondary interface. The primary interface will initiate RHPs by default. 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 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.

You can use two Ethernet interfaces.

NOTE
To take advantage of every interface in a Metro network, you can configure another MRP ring and either configure a different Master node for the ring or reverse the configuration of the primary and secondary interfaces on the Master node. Configuring multiple rings enables you to use all the ports in the ring. The same port can forward traffic one ring while blocking traffic for another ring.

Syntax: [no] enable

The **enable** command enables the ring.

Changing the hello and preforwarding times

You also can change the RHP hello time and preforwarding time. To do so, enter commands such as the following.

```
BigIron RX(config-vlan-2-mrp-1)# hello-time 200
BigIron RX(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.

NOTE
The preforwarding time must be at least twice the value of the hello time and must be a multiple of the hello time.

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, but 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 "MRP phase 2".

MRP phase 2

Beginning with the BIgIron RX release 02.4.00, Metro Ring Protocol (MRP) Phase 2 expands the functionality of MRP by allowing a physical interface that belong to the same VLAN to be shared by multiple rings.

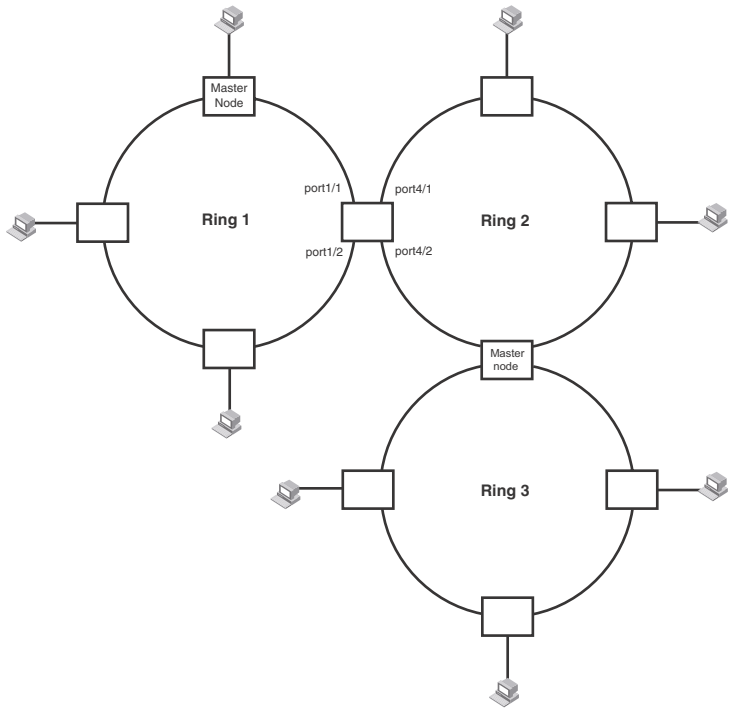
MRP is a Brocade proprietary protocol that prevents Layer 2 loops and provides fast reconvergence in Layer 2 ring topologies. It is an alternative to STP and is especially useful in Metropolitan Area Networks (MANs) that require more nodes and faster reconvergence time than what STP provides.

An MRP ring consists of nodes and each node has two interfaces on the ring. The interfaces on the ring must be members of the same port-based VLAN. Each node on the ring is connected to a separate customer network. The nodes forward Layer 2 traffic to and from the customer networks through the ring.

One node, is configured as the master node of the MRP ring. One of the two 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. The secondary interface blocks the packet to prevent a Layer 2 loops.

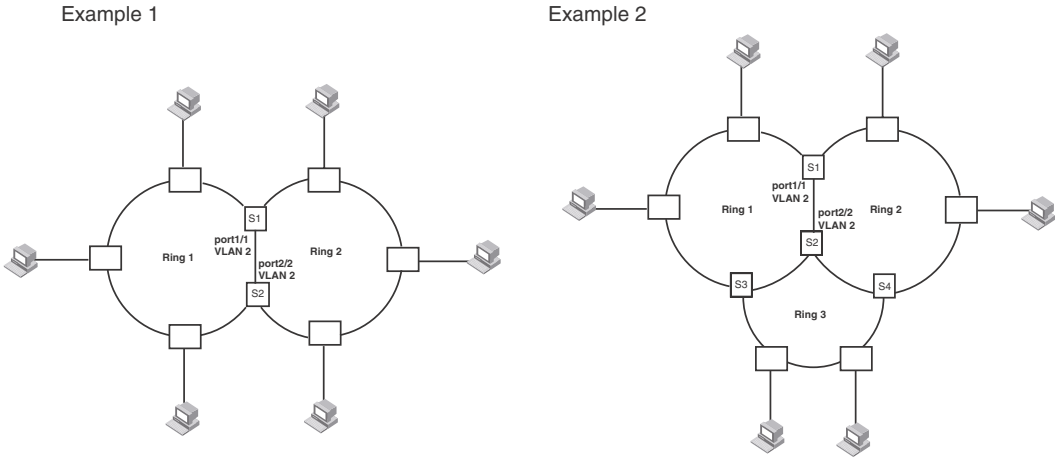
In MRP Phase 1, a node can have multiple MRP rings, but the rings cannot share the same interface. Also, when you configured an MRP ring, any node on the ring that is a BigIron Chassis device can be designated as the master node for the ring. Each ring is an independent ring and RHP packets are processed within each ring.

FIGURE 69 Multiple MRP rings - MRP phase 1



With MRP Phase 2, MRP rings can be configured to share the same interfaces as long as the interfaces belong to the same VLAN. Figure 69 shows examples of multiple MRP rings that share the same interface.

FIGURE 70 Examples of multiple rings sharing the same interface - MRP phase 2

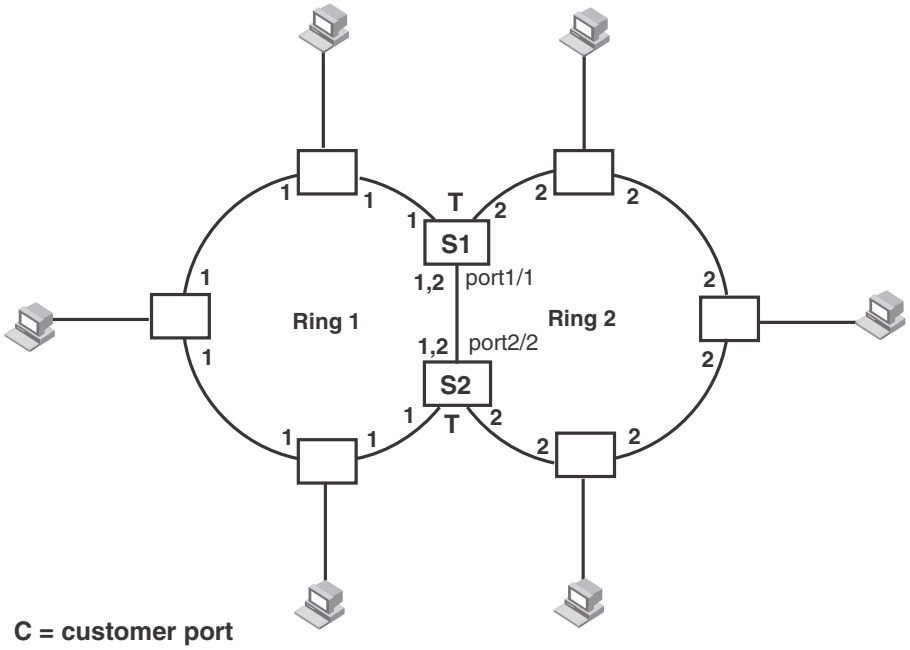


On each node that will participate in the ring, you specify the ring's ID and the interfaces that will be used for ring traffic. In a multiple ring configuration, a ring's ID determines its priority. The lower the ring ID, the higher priority of a ring.

A ring's ID is also used to identify the interfaces that belong to a ring.

Ring initialization for shared interfaces

FIGURE 71 Interface IDs and types



For example, in Figure 71, the ID of all interfaces on all nodes on Ring 1 is 1 and all interfaces on all nodes on Ring 2 is 2. Port 1/1 on node S1 and Port 2/2 on S2 have the IDs of 1 and 2 since the interfaces are shared by Rings 1 and 2.

The ring's ID is also used to determine an interface's priority. Generally, a ring's ID is also the ring's priority and the priority of all interfaces on that ring. However, if the interface is shared by two or more rings, then the highest priority (lowest ID) becomes the priority of the interface. For example, in Figure 71, all interfaces on Ring 1, except for Port 1/1 on node S1 and Port 2/2 on node S2 have a priority of 1. Likewise, all interfaces on Ring 2, except for Port 1/1 on node S1 and Port 2/2 on node S2 have a priority of 2. Port 1/1 on S1 and Port 2/2 on S2 have a priority of 1 since 1 is the highest priority (lowest ID) of the rings that share the interface.

If a node has interfaces that have different IDs, the interfaces that belong to the ring with the highest priority become regular ports. Those interfaces that do not belong to the ring with the highest priority become tunnel ports. In Figure 71, nodes S1 and S2 have interfaces that belong to Rings 1 and 2. Those interfaces with a priority of 1 are regular ports. The interfaces with a priority of 2 are the tunnel ports since they belong to Ring 2, which has a lower priority than Ring 1.

How ring breaks Are detected and healed between shared interfaces

If the link between ***shared interfaces*** breaks, the secondary interface on Ring 1's master node changes to a preforwarding state. The RHP packet sent by port 3/1 on Ring 2 is forwarded through the interfaces on S4, then to S2. The packet is then forwarded through S2 to S3, but not from S2 to S1 since the link between the two nodes is not available. When the packet reaches Ring 1's master node, the packet is forwarded through the secondary interface since it is currently in a preforwarding state. A secondary interface in preforwarding mode ignores any RHP packet that is not from its ring. The secondary interface changes to blocking mode only when the RHP packet forwarded by its primary interface is returned.

The packet then continues around Ring 1, through the interfaces on S1 to Ring 2 until it reaches Ring 2's master node. Port 3/2, the secondary interface on Ring 2 changes to blocking mode since it received its own packet, then blocks the packet to prevent a loop.

NOTE
On the ring member node, the primary and secondary interface is not only decided by the configuration, but also decide by the RHP flow from the ring master. The primary and secondary interface may not be swapped even if the configuration changes and there is an active ring master in the topology. If there is no active ring master in the topology, then the running configuration of the interface on the member node will follow what was configured.

Selection of master node

Allowing MRP rings to share interfaces limits the nodes that can be designated as the master node. Any node on an MRP ring that does not have a shared interface can be designated as the ring's master node. However, if all nodes on the ring have shared interfaces, nodes that do not have tunnel ports can be designated as the master node of that ring. If none of the nodes meet these criteria, you must change the rings' priorities by reconfiguring the rings' ID.

In Figure 71, any of the nodes on Ring 1, even S1 or S2, can be a master node since none of its interfaces are tunnel ports. However in Ring 2, neither S1 nor S2 can be a master node since these nodes contain tunnel ports.

RHP processing in rings with shared interfaces

Interfaces on an MRP ring have one of the following states:

- **Preforwarding (PF)** – All ring interfaces are in this state when you enable MRP.
- **Forwarding (F)** – An interface changes from Preforwarding to Forwarding when the port's preforwarding time expires.
- **Blocking (B)** – The interface cannot forward data. Only the secondary interface on the Master node can be Blocking.

The primary interface of the master node initiates the RHP packets and sends it on the ring. When the packet reaches an interface, MRP checks to see if the receiving interface is a regular port or a tunnel port.

If the port is a regular port, 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 blocks it.

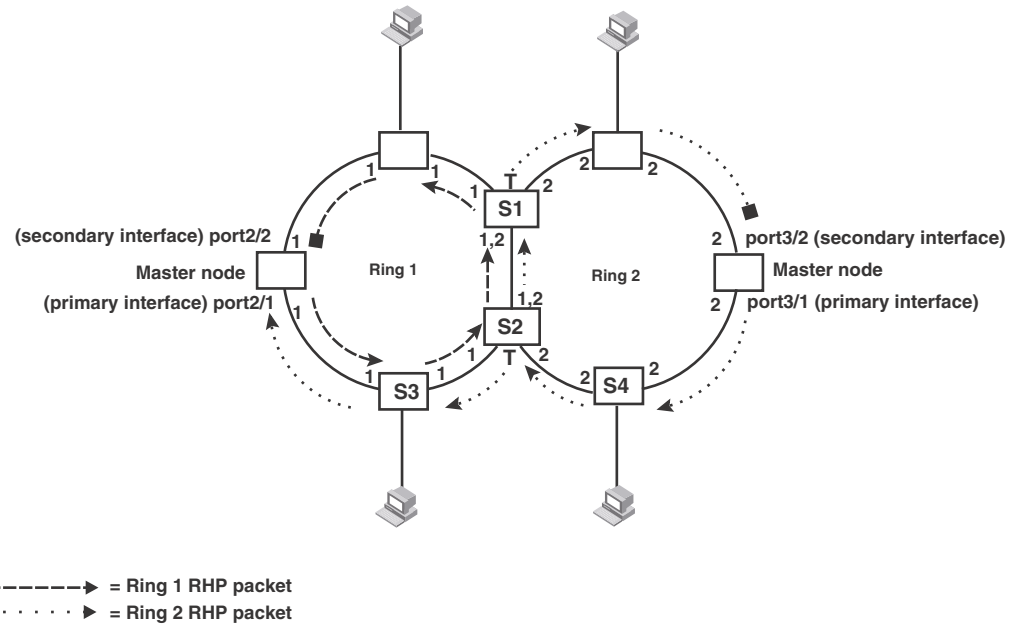
If the port is a tunnel port, MRP checks the priority of the RHP packet and compares it to the priority of the tunnel port:

- If the RHP packet's priority is less than or equal to the interface's priority, the packet is forwarded through that interface.
- If the priority of the RHP packet is greater than the priority of the interface, the RHP packet is dropped.

Normal flow

Figure 72 shows an example of how RHP packets are processed normally in MRP rings with shared interfaces.

FIGURE 72 Flow of RHP packets on MRP rings with shared interfaces



Port 2/1 on Ring 1's master node is the primary interface of the master node. The primary interface forwards an RHP packet on the ring. Since all the interfaces on Ring 1 are regular ports, the RHP packet is forwarded to all the interfaces until it reaches Port 2/2, the secondary interface of the master node. Port 2/2 then blocks the packet to complete the process.

On Ring 2, Port 3/1, is the primary interface of the master node. It sends an RHP packet on the ring. Since all ports on S4 are regular ports, the RHP packet is forwarded on those interfaces. When the packet reaches S2, the receiving interface is a tunnel port. The port compares the packet's priority to its priority. Since the packet's priority is the same as the tunnel port's priority, the packet is forwarded up the link shared by Rings 1 and 2.

When the RHP packet reaches the interface on node S2 shared by Rings 1 and 2, the packet is forwarded since its priority is less than the interface's priority. The packet continues to be forwarded to node S1 until it reaches the tunnel port on S1. That tunnel port determines that the RHP packet's priority is equal to the port's priority and forwards the packet. The RHP packet is forwarded to the remaining interfaces on Ring 2 until it reaches port 3/2, the secondary interface of the master node. Port 3/2 then blocks the packet to prevent a loop.

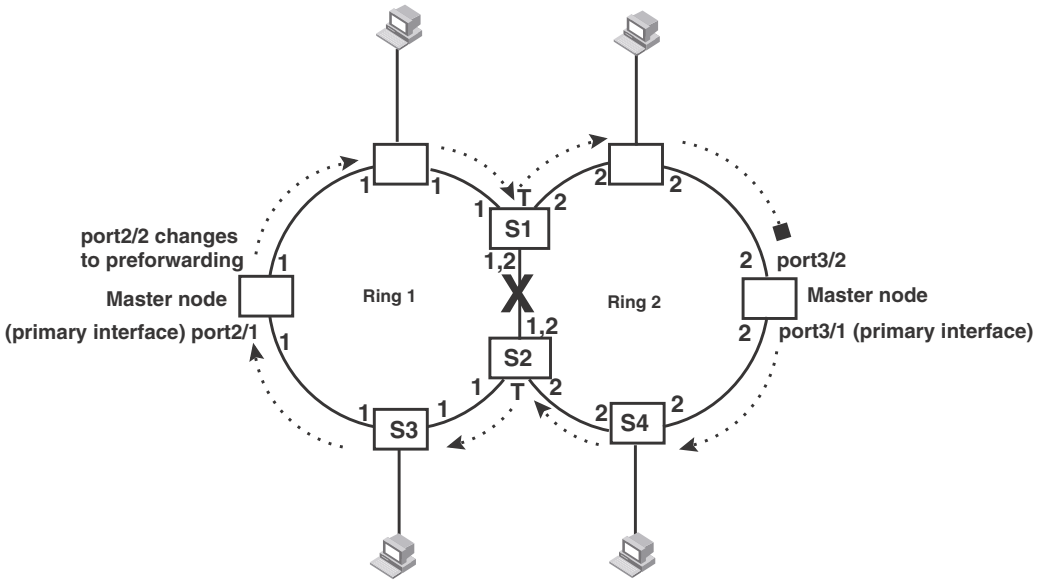
When the RHP packet from Ring 2 reached S2, it was also forwarded from S2 to S3 on Ring 1 since the port on S2 has a higher priority than the RHP packet. The packets is forwarded around Ring 1 until it reaches port 2/2, Ring 1's the secondary port. The RHP packet is then blocked by that port.

Flow when a link breaks

If the link between shared interfaces breaks (Figure 73), the secondary interface on Ring 1's master node changes to a preforwarding state. The RHP packet sent by port 3/1 on Ring 2 is forwarded through the interfaces on S4, then to S2. The packet is then forwarded through S2 to S3, but not from S2 to S1 since the link between the two nodes is not available. When the packet reaches Ring 1's master node, the packet is forwarded through the secondary interface since it is currently in a preforwarding state. A secondary interface in preforwarding mode ignores any RHP packet that is not from its ring. The secondary interface changes to blocking mode only when the RHP packet forwarded by its primary interface is returned.

The packet then continues around Ring 1, through the interfaces on S1 to Ring 2 until it reaches Ring 2's master node. Port 3/2, the secondary interface on Ring 2 changes to blocking mode since it received its own packet, then blocks the packet to prevent a loop.

FIGURE 73 Flow of RHP packets when a link for shared interfaces brakes



RHP packets follow this flow until the link is restored; then the RHP packet returns to it normal flow as shown in Figure 73.

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.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# metro-ring 1
BigIron RX(config-vlan-2-mrp-1)# name CustomerA
BigIron RX(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
BigIron RX(config-vlan-2-mrp-1)# enable
BigIron RX(config-vlan-2-mrp-1)# metro-ring 2
BigIron RX(config-vlan-2-mrp-2)# name CustomerB
BigIron RX(config-vlan-2-mrp-2)# ring-interface ethernet 1/1 ethernet 1/2
BigIron RX(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.

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.

```
BigIron RX(config-vlan-2-mrp-1)#diagnostics
```

Syntax: [no] diagnostics

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.

```
BigIron RX(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 79 CLI display of MRP ring diagnostic information

This field...	Displays...
Ring id	The ring ID.
Diag state	The state of ring diagnostics.
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 395.

Displaying MRP information

You can display the following MRP information:

- Topology group configuration information
- 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 438 for more information.

Displaying ring information

To display ring information, enter the following command.

```
BigIron RX(config)# show metro

Metro Ring 2
=============
Ring       State     Ring       Master     Topo       Hello      Prefwing
id                   role       vlan       group      time(ms)   time(ms)
2          enabled   member     2          not conf   100        300

Ring interfaces    Interface role     Forwarding state    Active interface
Interface Type
ethernet 1/1       primary            disabled            none
Regular
ethernet 1/2      secondary         forwarding         ethernet 2         Tunnel

RHPs sent          RHPs rcvd          TC RHPs rcvd        State changes
3                    0                  0                   4
```

Syntax: show metro [<*ring-id*>]

This display shows the following information.

TABLE 80 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
Master vlan	The ID of the master VLAN in the topology group used by this ring. If a topology group is used by MRP, the master VLAN controls the MRP settings for all VLANs in the topology group. **NOTE:** The topology group ID is 0 if the MRP VLAN is not the master VLAN in a topology group. Using a topology group for MRP configuration is optional.
Topo group	The topology group ID.
Hello time	The interval, in milliseconds, at which the Forwarding port on the ring's master node sends Ring Hello Packets (RHPs).

TABLE 80 CLI display of MRP ring information (Continued)

This field...	Displays...
Prefwing time	The number of milliseconds an MRP 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. **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.
Interface 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.
TC RHPs rcvd	The number of Topology Change RHPs received on the interface. A Topology Change RHP indicates that the ring topology has changed.
State changes	The number of MRP forwarding 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 68 on page 394.

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.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-2)# metro-ring 1
BigIron RX(config-vlan-2-mrp-1)# name "Metro A"
BigIron RX(config-vlan-2-mrp-1)# master
BigIron RX(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
BigIron RX(config-vlan-2-mrp-1)# enable
BigIron RX(config-vlan-2-mrp-1)# exit
BigIron RX(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.

```
BigIron RX(config)# vlan 30
BigIron RX(config-vlan-30)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-30)# tag ethernet 2/1
BigIron RX(config-vlan-30)# exit
BigIron RX(config)# vlan 40
BigIron RX(config-vlan-40)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-40)# tag ethernet 4/1
BigIron RX(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.

```
BigIron RX(config)# topology-group 1
BigIron RX(config-topo-group-1)# master-vlan 2
BigIron RX(config-topo-group-1)# member-vlan 30
BigIron RX(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.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-2)# metro-ring 1
BigIron RX(config-vlan-2-mrp-1)# name "Metro A"
BigIron RX(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
BigIron RX(config-vlan-2-mrp-1)# enable
BigIron RX(config-vlan-2)# exit
BigIron RX(config)# vlan 30
BigIron RX(config-vlan-30)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-30)# tag ethernet 2/1
BigIron RX(config-vlan-30)# exit
BigIron RX(config)# vlan 40
BigIron RX(config-vlan-40)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-40)# tag ethernet 4/1
BigIron RX(config-vlan-40)# exit
```

```
BigIron RX(config)# topology-group 1
BigIron RX(config-topo-group-1)# master-vlan 2
BigIron RX(config-topo-group-1)# member-vlan 30
BigIron RX(config-topo-group-1)# member-vlan 40
```

Commands on switch C

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-2)# metro-ring 1
BigIron RX(config-vlan-2-mrp-1)# name “Metro A”
BigIron RX(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
BigIron RX(config-vlan-2-mrp-1)# enable
BigIron RX(config-vlan-2)# exit
BigIron RX(config)# vlan 30
BigIron RX(config-vlan-30)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-30)# tag ethernet 2/1
BigIron RX(config-vlan-30)# exit
BigIron RX(config)# vlan 40
BigIron RX(config-vlan-40)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-40)# tag ethernet 4/1
BigIron RX(config-vlan-40)# exit
BigIron RX(config)# topology-group 1
BigIron RX(config-topo-group-1)# master-vlan 2
BigIron RX(config-topo-group-1)# member-vlan 30
BigIron RX(config-topo-group-1)# member-vlan 40
```

Commands on switch D

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-2)# metro-ring 1
BigIron RX(config-vlan-2-mrp-1)# name “Metro A”
BigIron RX(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
BigIron RX(config-vlan-2-mrp-1)# enable
BigIron RX(config-vlan-2)# exit
BigIron RX(config)# vlan 30
BigIron RX(config-vlan-30)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-30)# tag ethernet 2/1
BigIron RX(config-vlan-30)# exit
BigIron RX(config)# vlan 40
BigIron RX(config-vlan-40)# tag ethernet 1/1 to 1/2
BigIron RX(config-vlan-40)# tag ethernet 4/1
BigIron RX(config-vlan-40)# exit
BigIron RX(config)# topology-group 1
BigIron RX(config-topo-group-1)# master-vlan 2
BigIron RX(config-topo-group-1)# member-vlan 30
BigIron RX(config-topo-group-1)# member-vlan 40
```

Chapter

Virtual Switch Redundancy Protocol (VSRP) 15

In this chapter

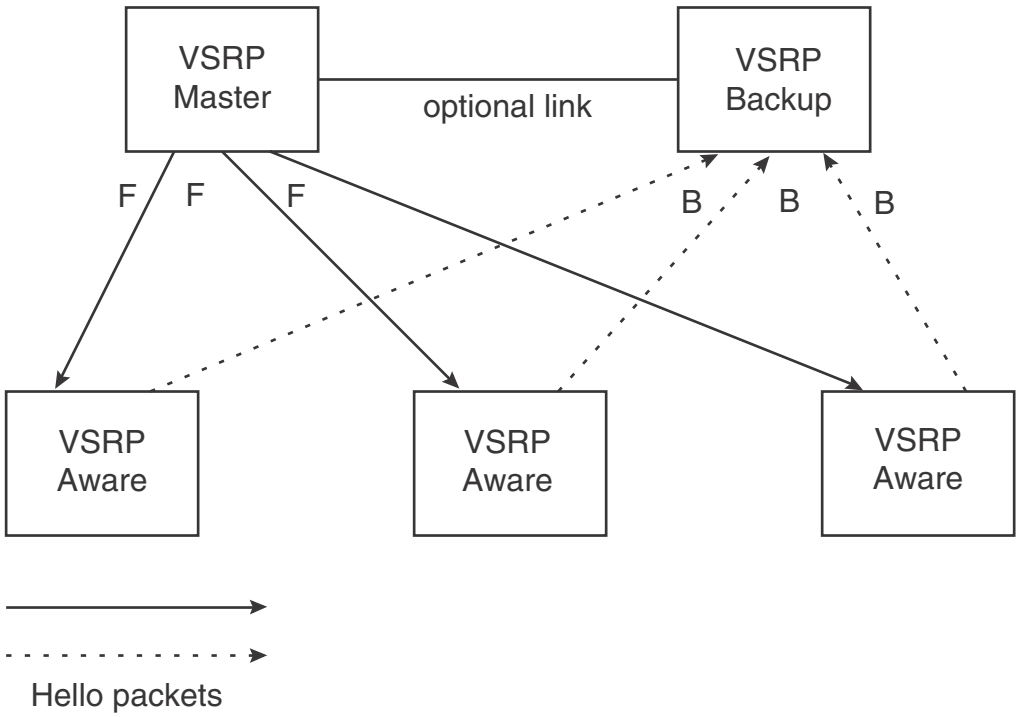
Overview of Virtual Switch Redundancy Protocol (VSRP)

VSRP is a Brocade proprietary protocol that provides redundancy and sub-second failover in Layer 2 and Layer 3 mesh topologies. Based on the Brocade'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 74 shows a VSRP configuration.

FIGURE 74 VSRP mesh – redundant paths for Layer 2 and Layer 3 traffic



In this example, two device devices are configured as redundant paths for VRID 1. On each device, 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 Brocade 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 Brocade devices can use the redundant paths provided by the VSRP devices. In this example, three Brocade devices use the redundant paths. A Brocade device that is not itself configured for VSRP but is connected to a Brocade device that is configured for VSRP, is ***VSRP aware***. In this example, the three Brocade devices connected to the VSRP devices are VSRP aware. A Brocade 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 Brocade devices connected to the VSRP devices has a separate link to each of the VSRP devices.

When using the device in conjunction with a FastIron Edge Switch, FastIron GS Series Switch, FastIron LS Series Switch, FastIron Edge Switch X Series Switch, or the FastIron Edge Switch X Series Switch as the VSRP-aware switches, the **vsrp-aware vrid <num> tc-vlan-flush** command is required to be configured on the non-BigIron RX devices. Refer to the *FastIron Configuration Guide* for additional information.

Layer 2 and Layer 3 redundancy

You can configure VSRP to provide redundancy for Layer 2 only or both for Layer 2 and Layer 3:

- **Layer 2 only** – The Layer 2 links are backed up but specific IP addresses are not backed up.
- **Layer 2 and Layer 3** – The Layer 2 links are backed 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.

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:

- The device whose virtual routing interface has a higher IP address becomes the master.
- If no IP address is configured, the device's base MAC address is used.

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.

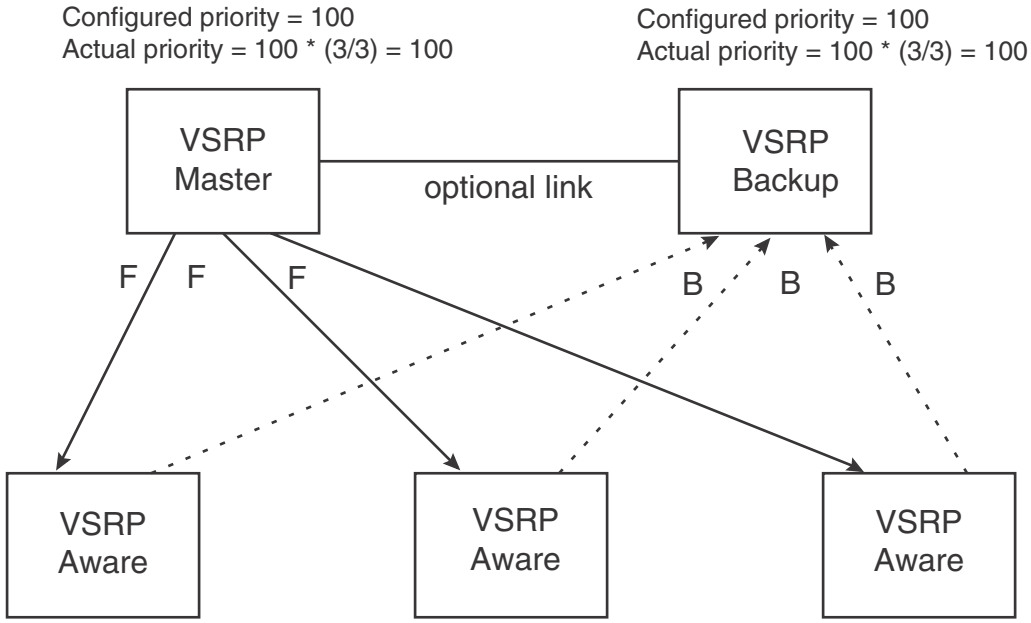
When a Backup sends a Hello message announcing its intent to become the Master, the Backup also starts a hold-down timer. During the hold-down time, the Backup listens for a Hello message with a higher priority than its own:

- 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.

VSRP priority calculation

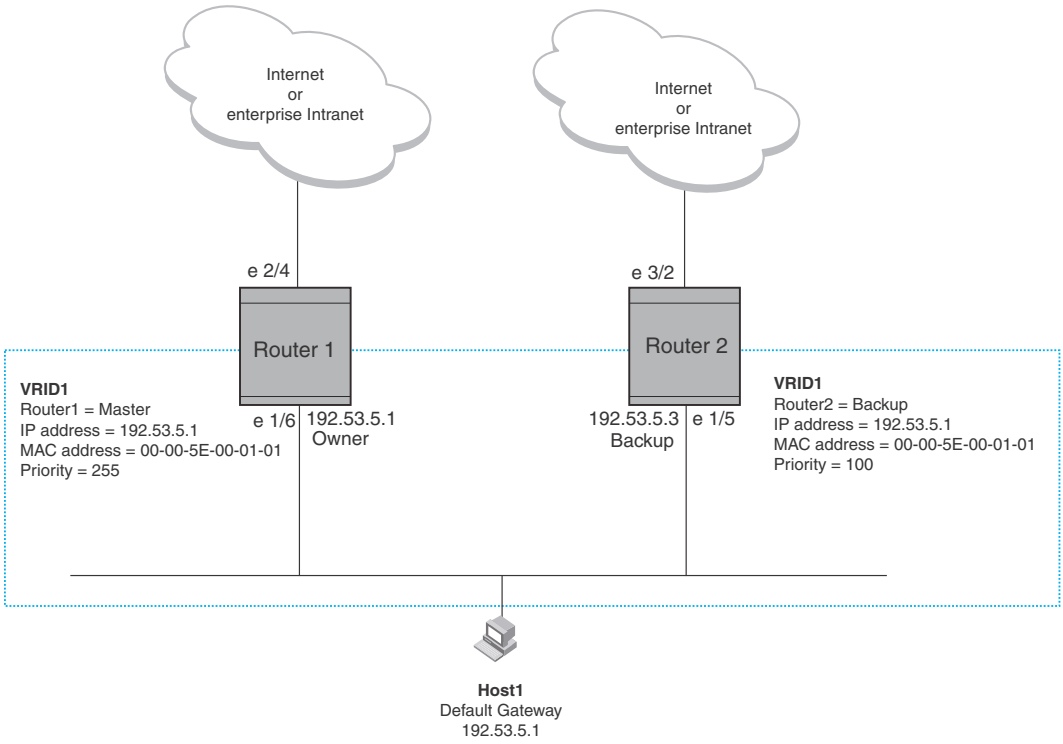
Each VSRP device has a VSRP priority for each VRID and its VLAN. The VRID is used during Master election for the VRID. By default, a device's VSRP priority is the value configured on the device (which is 100 by default). However, to ensure that a Backup with a high number of up ports for a given VRID 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 75

FIGURE 75 VSRP priority



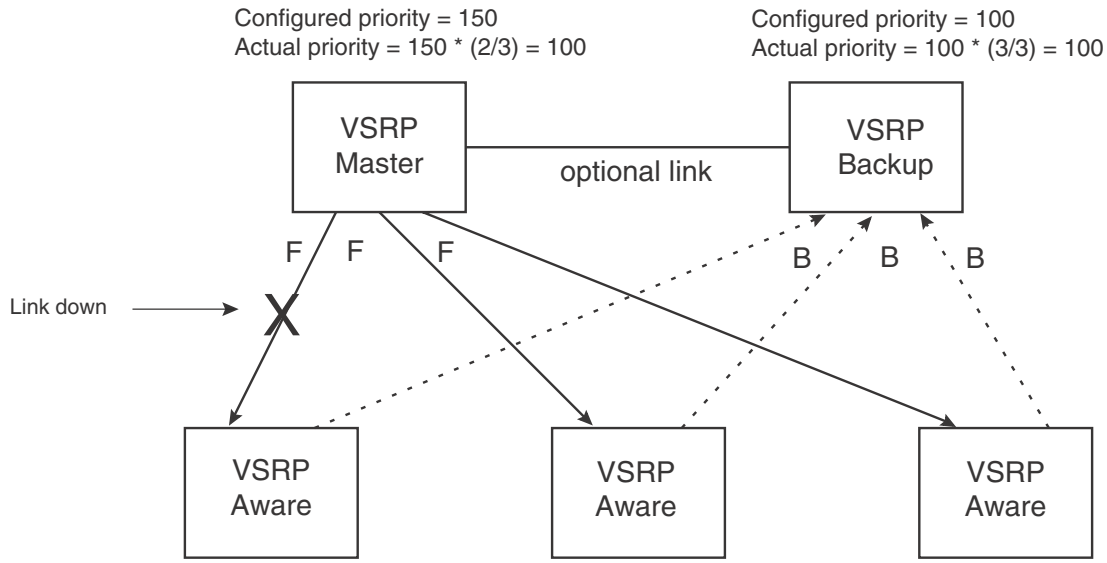
However, if one of the VRID's ports goes down on one of the Backups, that Backup's priority is reduced. If the Master's priority is reduced enough to make the priority lower than a Backup's priority, the VRID fails over to the Backup. Figure 76 shows an example.

FIGURE 76 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 76 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 77.

FIGURE 77 VSRP priority bias

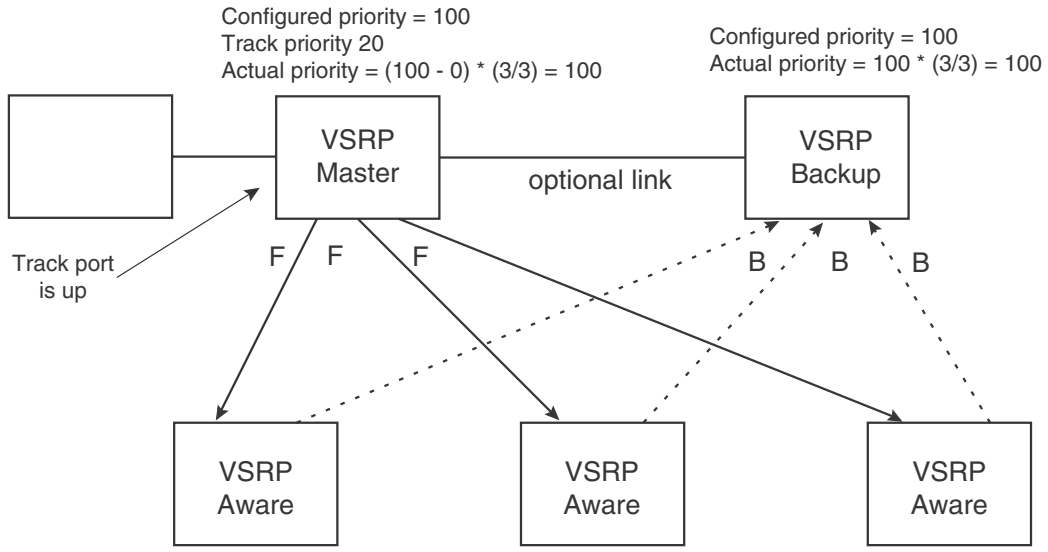


Track ports

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.

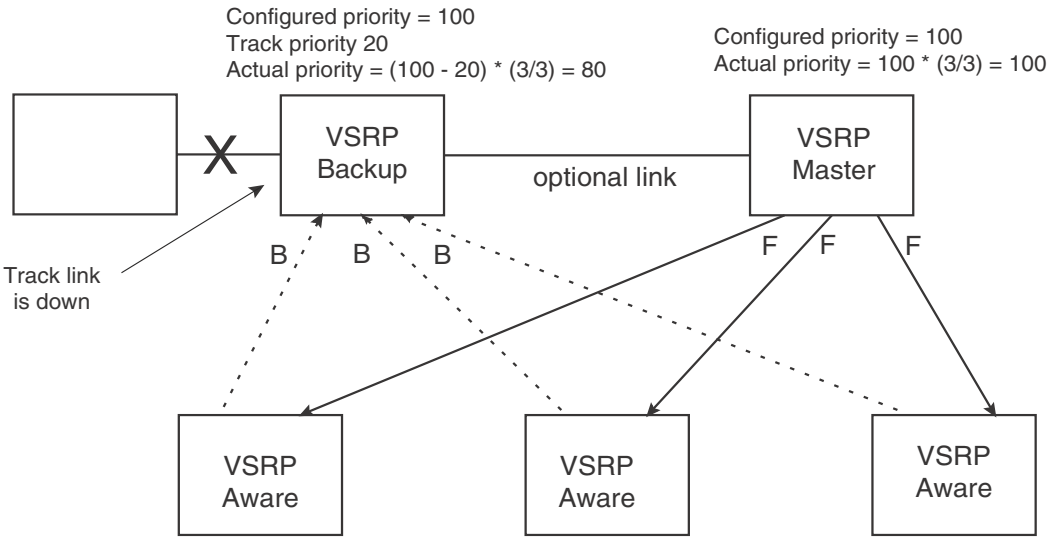
When you configure a track port, you assign a priority value to the port. If the port goes down, VSRP subtracts the track port's priority value from the configured VSRP priority. For example, if the you configure a track port with priority 20 and the configured VSRP priority is 100, the software subtracts 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 78 shows an example.

FIGURE 78 Track port priority



In Figure 78, 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 79.

FIGURE 79 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))/10

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 + 3 + (3 x 1)/10 = .9 seconds(900 milliseconds)

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.

1. 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 "Adding or removing a port from the VRID's VLAN" on page 420.

   ```
   BigIron RX(config)# vlan 200
   BigIron RX(config-vlan-200)# tag ethernet 1/1 to 1/8
   ```

2. Configure a VRID.

   ```
   BigIron RX(config-vlan-200)# vsrp vrid 1
   ```

   Syntax: [no] vsrp vrid <*num*>

   The <*num*> parameter specifies the VRID and can be from 1 – 255.

3. 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.

   ```
   BigIron RX(config-vlan-200-vrid-1)# backup
   ```

Syntax: [no] backup [priority <*value*>] [track-priority <*value*>]

The **backup** 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.

4. Enable VSRP on the VRID.

```
BigIron RX(config-vlan-200-vrid-1)# enable
```

Syntax: [no] enable

or

Syntax: [no] activate

For information about the command's optional parameters, see the following:

- "Changing the backup priority" on page 422
- "Changing the default track priority" on page 425

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.

```
BigIron RX(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.

Configuring optional VSRP parameters

The following sections describe how to configure optional VSRP parameters.

Disabling VSRP on a VRID

If you want to deactivate VSRP on a VRID, enter the following command.

```
BigIron RX(config-vlan-200-vrid-1)# disable
```

Syntax: disable

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.

```
BigIron RX(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.

Adding or 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 the port 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 an MRP ring.

To remove a port from a VRID, enter a command such as the following at the configuration level for the VRID.

```
BigIron RX(config-vlan-200-vrid-1)# no include-port ethernet 1/2
```

To return the port to the VRID, enter the following command.

```
BigIron RX(config-vlan-200-vrid-1)# 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.

Configuring a VRID IP address

If Layer 3 VSRP is enabled, you can specify an IP address to be backed up. When you specify an IP address, VSRP provides redundancy for that address. This is useful if you want to back up the gateway address used by hosts attached to the VSRP Backups.

VSRP does not require you to specify an IP address. If you do not specify an address, VSRP provides Layer 2 redundancy. If you do specify an address, VSRP provides Layer 2 and Layer 3 redundancy.

The Layer 3 redundancy support is the same as VRRPE support. For information, refer to Chapter 17, “Configuring VRRP and VRRPE”.

NOTE
The VRID IP address must be in the same subnet as a real IP address configured on the VSRP interface, but cannot be the same as a real IP address configured on the interface. Also, an IP address cannot be configured for a virtual routing interface.

NOTE
Failover applies to both Layer 2 and Layer 3.

To specify an IP address to back up, enter a command such as the following at the configuration level for the VRID.

```
BigIron RX(config-vlan-200-vrid-1)# ip-address 10.10.10.1
```

Syntax: [no] ip-address <*ip-addr*>

VSRP fast start

VSRP fast start allows non-Brocade or non-VSRP aware devices that are connected to a Brocade device that is the VSRP Master to quickly switchover 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.

Configuring VSRP fast start

The VSRP fast start feature can be enabled on a VSRP-configured Brocade 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.

```
FastIron(configure)#vlan 100
FastIron(configure-vlan-100)#vsrp vrid 1
FastIron(configure-vlan-100-vrid-1)#fast-start 5
```

Syntax: [no] fast-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.

```
FastIron(configure)#interface ethernet 1/1
FastIron(configure-if-1/1)#vsrp fast-start 5
```

Syntax: [no] vsrp fast-start <*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 the VSRP fast start feature enabled

The **show vsrp vrid** command shows the ports on which the VSRP fast start feature is enabled.

```
BigIron RX(config-vlan-10-vsrp-1)#sh vsrp
VLAN 10
 Auth-type no authentication
 VRID 1
 ========
  State         Administrative-status Advertise-backup Preempt-mode
Link-Redundancy
  Backup        Enabled              Disabled        True               Disabled
  Parameter        Configured Current    Unit/Formula
  Priority         100        100        (100-0)*(4.0/4.0)
  Hello-interval   1          1          sec/10
  Hold-interval    3          3          sec/10
  Initial-ttl       2          2          hops

  Master router 219.218.18.52 or MAC xxxx.dbda.1234 expires in 00:00:02
  Member ports:      ethe 19/1 to 19/2 ethe 19/4 to 19/5
  Operational ports: ethe 19/1 to 19/2 ethe 19/4 to 19/5
  Forwarding ports:  None
  fast-start ports:     19/1(10) 19/2(10) 19/4(10) 19/5(1)
  Track-port 19/3 priority 50 status up
```

The "fast-start ports:" line lists the ports that have the VSRP fast start enabled, and the downtime for each port.

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 414 and "Changing the default track priority" on page 425.

To change the backup priority, enter a command such as the following at the configuration level for the VRID.

```
BigIron RX(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 425.

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

Each Backup saves the configured timer values to its startup configuration file when you save the device's configuration.

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.

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 3 seconds, enter the following command.

```
BigIron RX(config)# router vsrp
BigIron RX(config-vsrp-router)# slow-start 30
```

Syntax: 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, 3 seconds) before the Master takes over from the Backup.

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 Layer 3 Switch 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
An 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.

```
BigIron RX(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.

```
BigIron RX(config-vlan-200-vrid-1)# hello-interval 10
```

Syntax: [no] hello-interval <*units*>

The <*units*> parameter specifies the interval which and can be from 1 – 84 units. The default is 1 (1 unit = 100 milliseconds).

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 milliseconds a Backup waits for a Hello message from the Master before determining that the Master is dead. The default is 300 milliseconds. 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.

```
BigIron RX(config-vlan-200-vrid-1)# dead-interval 30
```

Syntax: [no] dead-interval <*units*>

The <*units*> parameter specifies the interval which and can be from 3 – 84 units (1 unit = 100 milliseconds). The default is 3 (3 units = 300 milliseconds).

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.

```
BigIron RX(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 6 seconds by default. You can change the interval to be up to 360 seconds.

To change the Backup Hello interval, enter a command such as the following at the configuration level for the VRID.

```
BigIron RX(config-vlan-200-vrid-1)# backup-hello-interval 180
```

Syntax: [no] backup-hello-interval <*units*>

The <*units*> parameter specifies the message interval and can be from 60 – 3600 units (1 unit = 100 milliseconds). The default is 60 units (6000 milliseconds or 6 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.

```
BigIron RX(config-vlan-200-vrid-1)# hold-down-interval 4
```

Syntax: [no] hold-down-interval <*units*>

The <*units*> parameter specifies the hold-down interval and can be from 2 – 84 units (1 unit = 100 milliseconds). The default is 3 (3 units = 300 milliseconds)

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 426.

To change the track priority, enter a command such as the following at the configuration level for the VRID.

```
BigIron RX(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 414.

To configure a VRID to track an interface, enter a command such as the following at the configuration level for the VRID.

```
BigIron RX(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 pre-emption

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.

```
BigIron RX(config-vlan-200-vrid-1)# non-preempt-mode
```

Syntax: [no] non-preempt-mode

Port transition hold timer

Using this VSRP command will delay the sending of port "up"/"down" events. 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 would 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 trunk 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.

```
BigIron RX (config-if-e1000-1/2)# delay-link-event 2 down
```

Syntax: delay-link-event <*time*> up | down

The <*time*> parameter is the number of 50-ms units. The default is 0.

The <*up*> parameter means only "up" events are delayed.

The <*down*> parameter means that only the down events are delayed.

Clearing VSRP information

You can clear all VSRP statistics, globally and per-instance, by entering the following command.

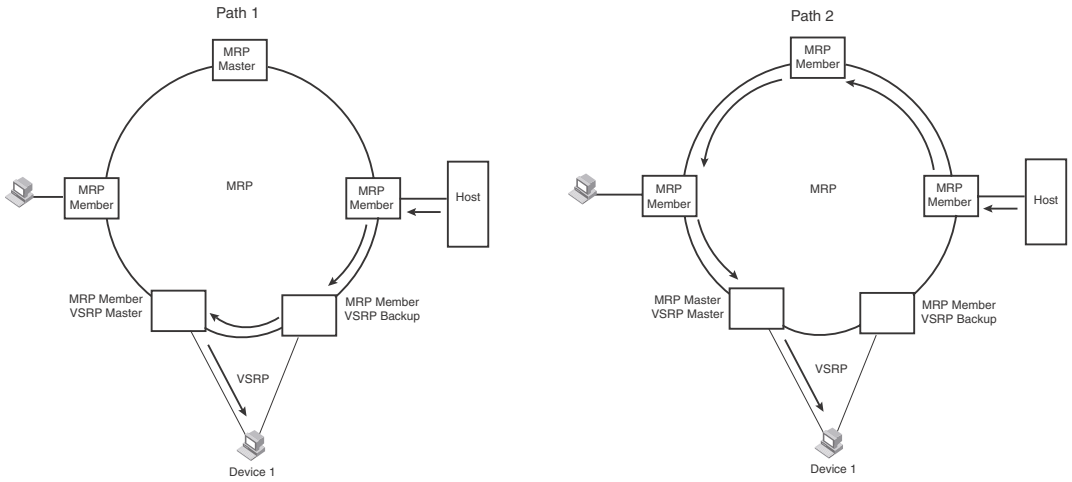
```
BigIron RX# clear vsrp
```

Syntax: clear vsrp

VSRP and MRP signaling

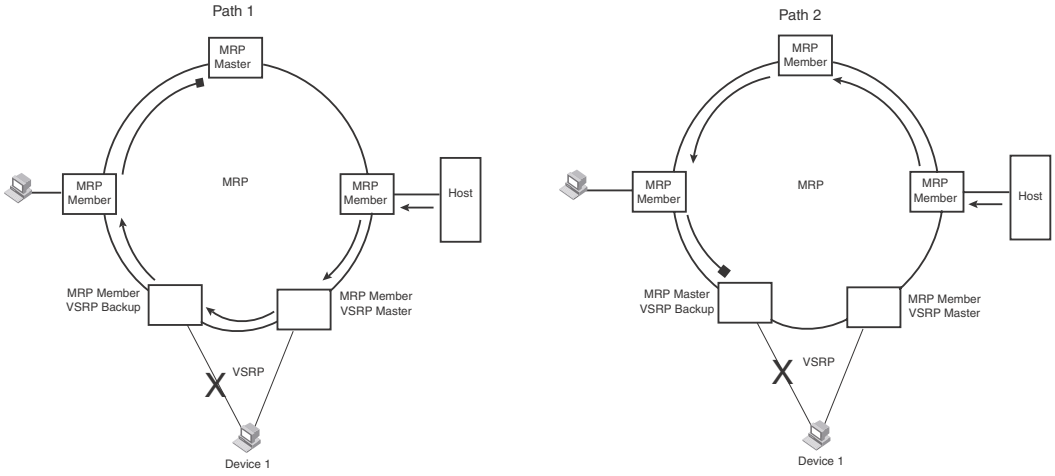
A device may connect to an MRP ring through VSRP to provide a redundant path between the device and the MRP ring. VSRP and MRP signaling, ensures rapid failover by flushing MAC addresses appropriately. The host on the MRP ring learns the MAC addresses of all devices on the MRP ring and VSRP link. From 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 80 below shows two possible data paths from the host to Device 1.

FIGURE 80 Two data paths from host on an MRP ring to a VSRP-linked device



If a VSRP failover from master to backup occurs, VSRP needs to inform 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 81.

FIGURE 81 VSRP on MRP rings that failed over

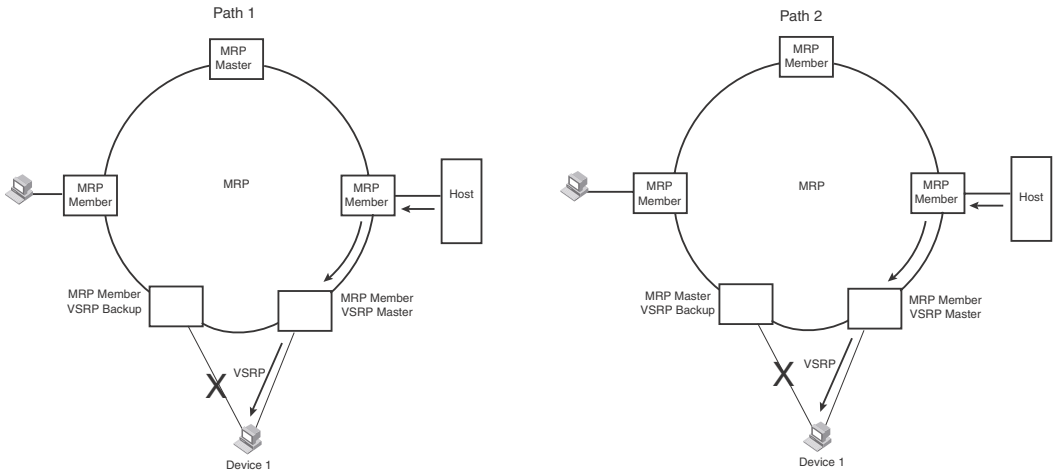


To ensure that MRP is informed of the topology change and to achieve convergence rapidly, there is a signaling process that controls the interaction between VSRP and MRP. When a VSRP node fails, a new VSRP master is selected. The new VSRP master finds all MRP instances impacted by the failover. Then each MRP instance does the following:

- The MRP node sends out an MRP PDU with the mac-flush flag set three times on the MRP ring.
- The MRP node that receives this MRP PDU empties all the MAC address entries from its interfaces that participate on the MRP ring.
- The MRP node then forwards the MRP PDU with the mac-flush flag set to the next MRP node that is in forwarding state.

The process continues until the Master 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 82).

FIGURE 82 New path established



There are no CLI commands used to configure this process.

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 detailed VSRP information, enter the **show vsrp vrid** or **show vsrp vlan** command. Both commands show the same information as in the following example.

```
BigIron RX# show vsrp vrid 100
VLAN 10
Auth-type no authentication
VRID 10
========
State Administrative-status Advertise-backup Preempt-mode
Master Enabled Disabled True
Parameter Configured Current Unit/Formula
Priority 100 80 (100-0)*(4.0/5.0)
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 60 60 sec/10
Next hello sent in 00:00:01
Member ports: ethe 3/1 ethe 3/3 ethe 6/1 ethe 11/1 ethe 14/1
Operational ports: ethe 3/1 ethe 3/3 ethe 6/1 ethe 14/1
Forwarding ports: ethe 3/1 ethe 3/3 ethe 6/1 ethe 14/1
Restart ports: 3/1(10) 3/3(10) 6/1(10) 11/1(10) 14/1(10)
```

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 433.

TABLE 81 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.

TABLE 81 CLI display of VSRP VRID or VLAN information (Continued)

This field...	Displays...
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 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 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.
save-current	The source of VSRP timer values preferred when you save the configuration. This field can have one of the following values: • **false** – The timer values configured on this device are saved. • **true** – The timer values most recently received from the Master are saved instead of the locally configured values.
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 units between Hello messages from the Master to the Backups for a given VRID (1 unit = 100 milliseconds).

TABLE 81 CLI display of VSRP VRID or VLAN information (Continued)

This field...	Displays...
dead-interval	The configured value for the dead interval. The dead interval is the number of units a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active (1 unit = 100 milliseconds). 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:** The value is never 0, as it defaults to 3 units.
hold-interval	The number of units a Backup that intends to become the Master will wait before actually beginning to forward Layer 2 traffic for the VRID (1 unit = 100 milliseconds). 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:** An 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 Layer 3 Switch 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.
Restart ports	Lists the ports on which VSRP fast start enabled.

Displaying a summary of VSRP information

To obtain a summary of VSRP Information, enter the **show vsrp brief** command. If the command is entered on a VSRP Backup, it displays the following information.

```
BigIron RX# show vsrp brief
VLAN VRID ConfPri CurPri P  State  PeerMacAddruir  IpAddress      VIP
  10   10     100     80 P  Backup  xxxx.db21.11a0  219.33.17.160  None
```

When the command is entered on a VSRP Master, it displays the following information.

```
BigIron RX# show vsrp brief
VLAN VRID ConfPri CurPri P  State  PeerMacAddr or IpAddress  VIP
  10   10     100     80 P  Master     Unknown      Unknown  None
```

When the command is entered on a Layer 3 VSRP, it displays the following information.

```
BigIron RX# show vsrp brief
VLAN VRID ConfPri CurPri P State  PeerMacAddr or IpAddress      VIP
```

```
100  1    150     1      P Initia xxxx.1414.1404 20.20.20.4     20.20.20.100
101  2    50      1     P Initia xxxx.1e1e.1e01 30.30.30.1     30.30.30.100
```

Syntax: show vsrp brief

This field...	Displays...
VLAN	The VLAN on which VSRP is configured.
VRID	The VRID for which the following information is displayed.
ConfPri	The configured priority for the device's preferability for becoming the Master for the VRID.
CurPri	The device's current priority for becoming the Master.
P	Pre-empt mode status: • **P** – pre-emption is enabled for the VLAN • **N** – pre-emption is disabled for the VLAN
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 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 VRID. • **standby** – This device is a Backup for the VRID. • **master** – This device is the Master for the VRID.
Peer MAC address or IP address	MAC address or IP address of the peer.
VIP	Virtual IP address for the VLAN.

Displaying VSRP packet statistics for VSRP

When Layer 3 VSRP is enabled, you can enter the following command to display VSRP statistics.

```
BigIron RX# show vsrp statistics
Global VSRP statistics
----------------------
- received packets with checksum errors = 0
- received packets with invalid version number = 0
- received packets with unknown or inactive vrid = 43
```

```
To display VSRP statistics for a specific VLAN, enter a command such as the
following:
BigIron RX# show vsrp statistics vlan 100
```

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) using the **show vsrp aware** command. The command shows the active interfaces for the VRID. No output is displayed if the command is entered on a VSRP master or backup.

```
BigIron RX# 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 429.

TABLE 82 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.

Chapter

16

Topology Groups

In this chapter

Topology overview

This chapter describes the different types of topology groups and how to configure them. 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 device is deployed in a Metro network and provides forwarding for two 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
- 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:

- **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 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.
- **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 MRP

A topology group enables you to control forwarding in multiple VLANs using a single instance of a Layer 2 protocol such as MRP. For more information on topology group and MRP, refer to "Master VLANs and customer VLANs in a topology group" on page 394.

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 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
Since free ports are not controlled by the master port's Layer 2 protocol, they are assumed to always be in the Forwarding state when enabled.

Configuration considerations

- You can configure up to 256 topology groups. Each group can control up to 4094 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, configure the master VLAN and member VLANs or member VLAN groups before you configure a topology group.
- Once you add a VLAN as a member of a topology group, all the Layer 2 protocol information on the VLAN is deleted.

- 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. For example, if you remove master VLAN 2, the software converts member VLAN 3 into the new master VLAN. The new master VLAN inherits the Layer 2 protocol settings of the older master VLAN.
- Once 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. Once 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 will need to reconfigure the Layer 2 protocol information in the VLAN or VLAN group.

Configuring a topology group

To configure a topology group, enter commands such as the following.

```
BigIron RX(config)# topology-group 2
BigIron RX(config-topo-group-2)# master-vlan 2
BigIron RX(config-topo-group-2)# member-vlan 3
BigIron RX(config-topo-group-2)# member-vlan 4
BigIron RX(config-topo-group-2)# member-vlan 5
BigIron RX(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 – 256 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.

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.

Displaying topology group information

The following sections show how to display topology group information for VLANS.

Displaying topology group information

To display topology group information, enter the following command.

```
BigIron RX(config)# show topology-group

Topology Group 1
    ==================
    Master VLAN   : 2
    Member VLAN   : 10 20 30
    Member Group  : None
    Control Ports : ethe 2/2 ethe 3/18 ethe 4/1 to 4/2
    Free Ports :
    Topology Group 2
    ==================
    Master VLAN   : 3
    Member VLAN   : 100 200
    Member Group  : None
    Control Ports : ethe 4/1 to 4/2
    Free Ports :
    VLAN 2    - ethe 2/1 ethe 3/17
    VLAN 10   - ethe 2/1 ethe 3/17
    VLAN 20   - ethe 2/1 ethe 3/17
    VLAN 30   - ethe 2/1 ethe 3/17
```

Syntax: show topology-group [<*group-id*>]

This display shows the following information.

TABLE 83 CLI display of topology group information

This field...	Displays...
master-vlan	The master VLAN for the topology group. The settings for STP, 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.
member-vlan	The member VLANs in the topology group.
Common 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.
L2 protocol	The Layer 2 protocol configured on the control ports. The Layer 2 protocol can be one of the following: • MRP • STP • RSTP • VSRP
Per vlan free ports	The ports that are not controlled by the Layer 2 protocol information in the master VLAN.

Chapter

17

Configuring VRRP and VRRPE

In this chapter

Overview of VRRP

This chapter describes how to configure the following router redundancy protocols:

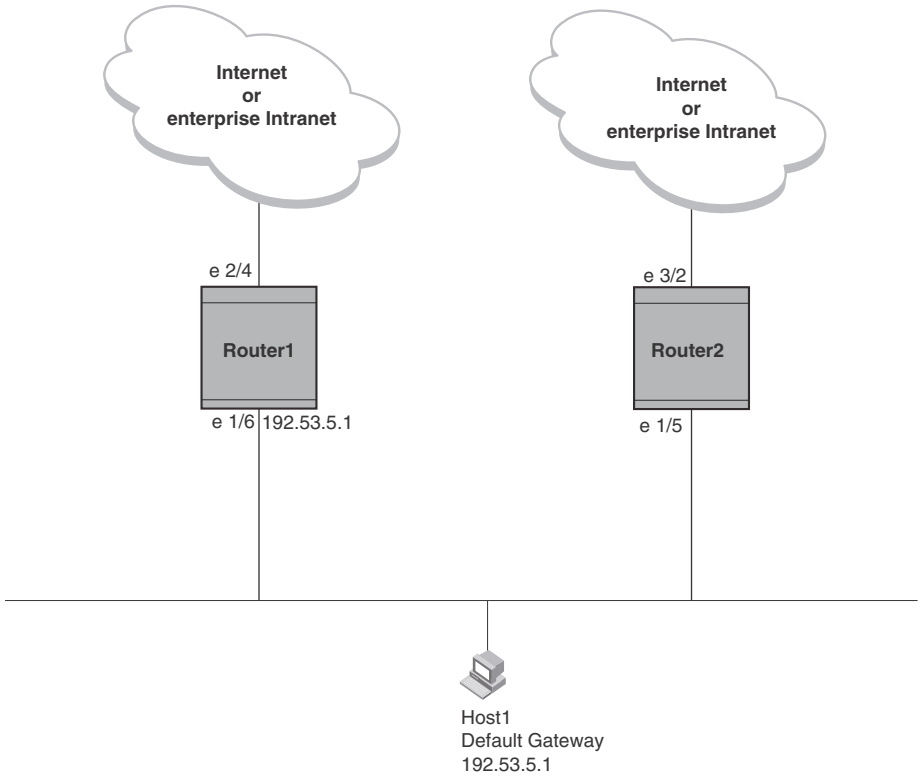
- ***Virtual Router Redundancy Protocol (VRRP)*** – The standard router redundancy protocol described in RFC 3768.
- ***VRRP Extended (VRRPE)*** – A Brocade proprietary version of VRRP that overcomes limitations in the standard protocol. This protocol works only with Brocade devices.

This section presents the standard VRRP options and the options that Brocade 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 83.

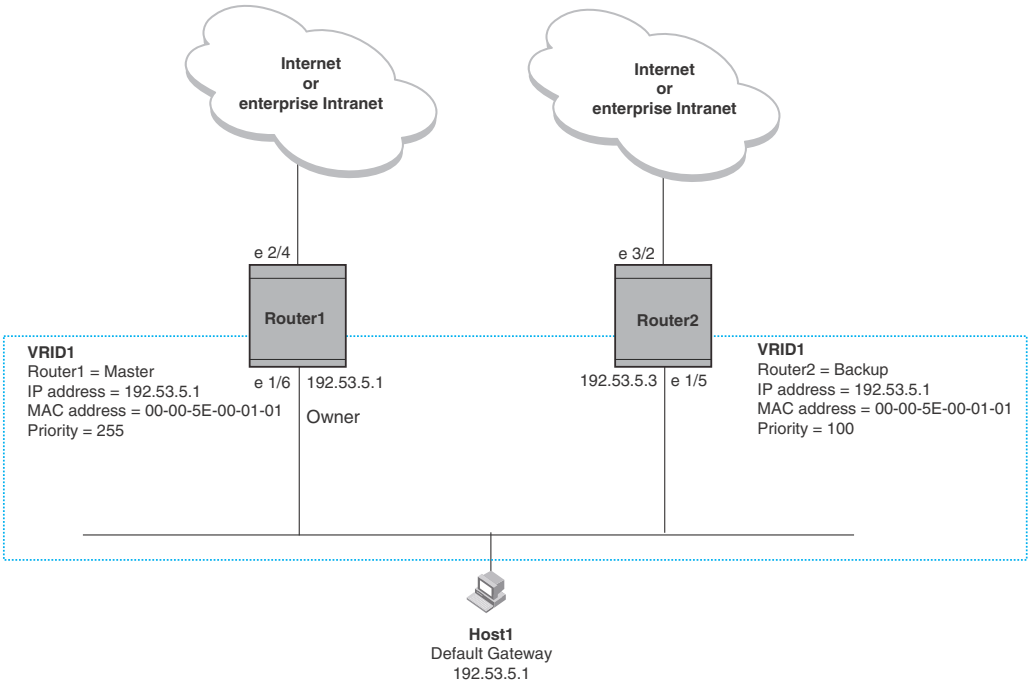
FIGURE 83 Router1 is Host1's default gateway but is a single point of failure



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 84, Router 2 provides the default gateway out of the subnet if Router 1 fails.

FIGURE 84 Router1 and Router2 are configured as a VRRP virtual router to provide redundant 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 84 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 84, 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 device 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 3 – 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 84, 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).

Brocade enhancements of VRRP

Brocade enhanced VRRP by adding the following options:

- Track Ports and Track Priority
- Suppression of RIP Advertisements for Backed Up Interfaces
- Authentication
- VRRP's operation is independent of RIP, OSPF, and BGP

Track ports and track priority

Brocade 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 84 on page 441, 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 84 on page 441, 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 84 on page 441, 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 address(es) 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 address(es) than the track priority you assign on the Backup routers.

Suppression of RIP advertisements for backed up interfaces

The Brocade 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 Brocade 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 for VRRP.

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 VRRPE

VRRPE is Brocade's proprietary version of VRRP that overcomes limitations in the standard protocol. It is similar to VRRP, but differs in the following respects:

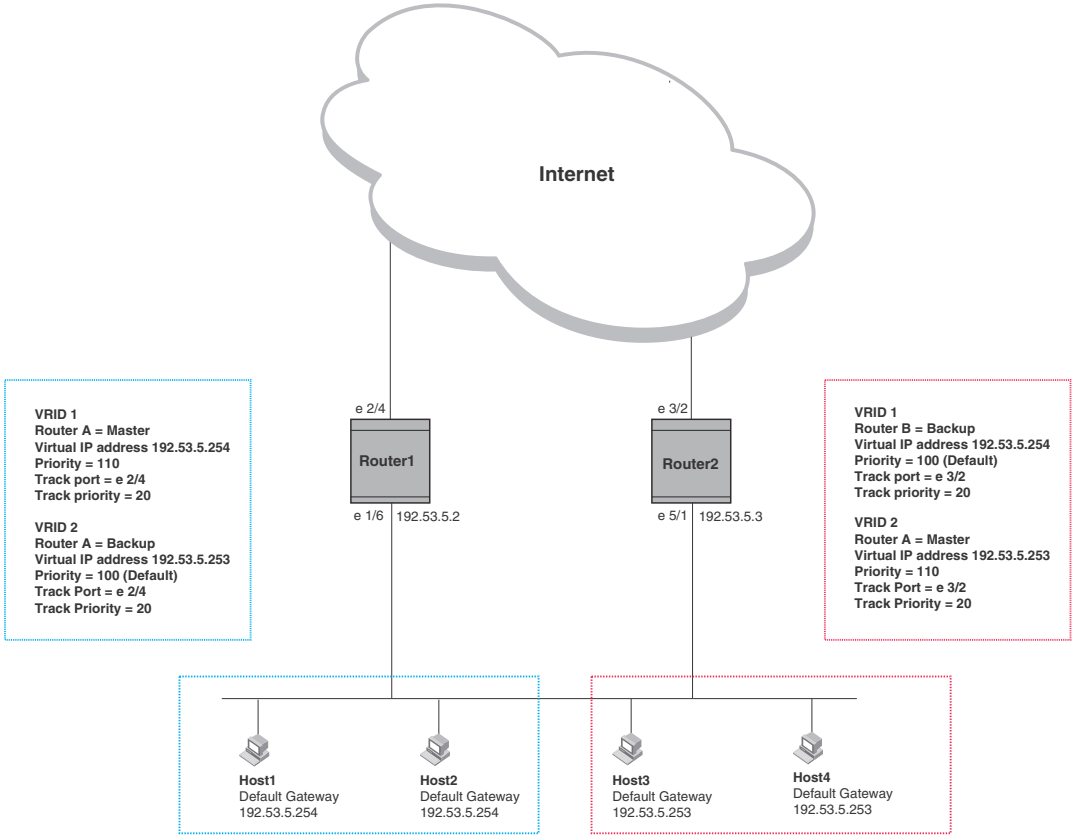
- Owners and Backup:
  - VRRP has an Owner and one or more Backups for each virtual router. The Owner is the router that has the IP address used for the virtual router. All the other routers supporting the virtual router are Backups.
  - VRRPE does not use Owners. All routers are Backups for a given 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.
  - VRRPE – The Master and Backups are selected based on their priority. You can configure any of the device 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.
  - VRRPE 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, VRRPE does not allow you to specify an IP address configured on the interface as the VRID IP address.
- VRID's MAC Address:
  - VRRP source MAC is a virtual MAC address defined as 00-00-5E-00-01-<vrid>, where <vrid> is the ID of the virtual router. The Master owns the Virtual MAC address.
  - VRRPE uses the interface's actual MAC address as the source MAC address. The virtual MAC address is 02-E0-52-<hash-value>-<vrid>, where <hash-value> is a two-octet hashed value for the IP address and <vrid> is the VRID.
- Hello packets:
  - VRRP sends Hello messages to IP Multicast address 224.0.0.18.
  - VRRPE 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.
  - VRRPE reduces the priority of a VRRPE interface by the amount of a tracked interface's priority if the tracked interface's link goes down. For example, if the VRRPE interface's priority is 200 and a tracked interface with track priority 20 goes down, the software changes the VRRPE 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 VRRPE routers are Backups. There is no Owner router. VRRPE overcomes the limitations in standard VRRP by removing the Owner.

Figure 85 shows an example of a VRRPE configuration.

FIGURE 85 Router1 and Router2 are configured to provide dual redundant network access for the host



In this example, Router1 and Router2 use VRRPE to load share as well as provide redundancy to the hosts. The load sharing is accomplished by creating two VRRPE 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 device configured for VRRPE can interoperate only with other device.

VRRP and VRRPE parameters

Table 84 lists the VRRP and VRRPE parameters. Most of the parameters and default values are the same for both protocols. The exceptions are noted in the table.

TABLE 84 VRRP and VRRPE parameters

Parameter	Description	Default	See page...
Protocol	The Virtual Router Redundancy Protocol (VRRP) based on RFC 2338 or VRRP-Extended, Brocade's enhanced implementation of VRRP	Disabled **NOTE:** Only one of the protocols can be enabled at a time.	page 448 page 450
VRRP or VRRPE router	The device's active participation as a VRRP or VRRPE router. Enabling the protocol does not activate the device for VRRP or VRRPE. You must activate the device as a VRRP or VRRPE router after you configure the VRRP or VRRPE parameters.	Inactive	page 448 page 450
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 448 page 450
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. • **VRRPE** – The virtual router IP address must be in the same subnet as a real IP address configured on the VRRPE interface, but cannot be the same as a real IP address configured on the interface.	None	page 448 page 450
VRID MAC address	The source MAC address in VRRP or VRRPE 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>. The Master owns the Virtual MAC address. • **VRRPE** – A virtual MAC address defined as 02-E0-52-<hash-value>-<vrid>, where <hash-value> is a two-octet hashed value for the IP address and <vrid> is the VRID.	Not configurable	page 442

TABLE 84 VRRP and VRRPE parameters (Continued)

Parameter	Description	Default	See page...
Authentication type	The type of authentication the VRRP or VRRPE routers use to validate VRRP or VRRPE 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 VRRPE.	No authentication	page 443 page 451
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. VRRPE – All routers for the VRID are Backups.	page 448 page 450
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 – 254. • **VRRPE** – 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 VRRPE – 100 for all Backups	page 448 page 450
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 452
Hello interval	The number of seconds between Hello messages from the Master to the Backups for a given VRID. The interval can from 1 – 84 seconds.	One second	page 452
Dead interval	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.	Three times the Hello Interval plus one-half second	page 453
Backup Hello interval	The number of seconds between Hello messages from a Backup to the Master. The message interval can be from 60 – 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 453

TABLE 84 VRRP and VRRPE parameters (Continued)

Parameter	Description	Default	See page...
Track port	Another device 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 VRRPE priority of the VRID interface is changed, causing the devices to renegotiate for Master.	None	page 442 page 453
Track priority	A VRRP or VRRPE priority value assigned to the tracked ports. If a tracked port's link goes down, the VRID port's VRRP or VRRPE priority changes. • **VRRP** – The priority changes to the value of the tracked port's priority. • **VRRPE** – The VRID port's priority is reduced by the amount of the tracked port's priority.	VRRP – 2 VRRPE – 5	page 442 page 454
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 454

Configuring parameters specific to VRRP

VRRP is configured at the interface level. To implement a simple VRRP configuration using all the default values, enter commands such as the following.

Configuring the owner

To configure the VRRP Owner router, enter the following commands on the router that will be the Owner.

```
Router1(config)# router vrrp
Router1(config)# inter e 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
Router1(config-if-e10000-1/6-vrid-1)# ip-address 192.53.5.1
Router1(config-if-e10000-1/6-vrid-1)# activate
```

Syntax: router vrrp

Syntax: ip vrrp vrid <*num*>

Syntax: owner [track-priority <*value*>]

Syntax: activate

The **track-priority** <*value*> parameter changes the track-port priority for this interface and VRID from the default (2) to a value from 1 – 254.

Syntax: ip-address <*ip-addr*>

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 for VRRP" on page 449 for additional requirements.

Configuring basic VRRP parameters

To implement a simple VRRP configuration using all the default values, enter commands such as the following.

Configuring the owner

```
Router1(config)# router vrrp
Router1(config)# inter e 1/6
Router1(config-if-1/6)# ip address 192.53.5.1
Router1(config-if-1/6)# ip vrrp vrid 1
Router1(config-if-1/6-vrid-1)# owner
Router1(config-if-1/6-vrid-1)# ip-address 192.53.5.1
Router1(config-if-1/6-vrid-1)# activate
```

Configuring a backup

To configure the VRRP Backup router, enter the following commands.

```
Router2(config)# router vrrp
Router2(config)# inter e 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
Router2(config-if-1/5-vrid-1)# advertise backup
Router2(config-if-e10000-1/5-vrid-1)# ip-address 192.53.5.1
Router2(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: router vrrp

Syntax: backup [priority <value>] [track-priority <*value*>]

The **priority** <value> parameter specifies the VRRP priority for this virtual router. You can specify a value from 3 – 254. The default is 100.

Enter a value of 3 – 254 for the **track-priority** <value> parameter if you want VRRP to monitor the state of the interface. The default is 100.

Syntax: ip-address <*ip-addr*>

Refer to "Configuration rules for VRRP" on page 449 for additional requirements.

Configuration rules for VRRP

- The interfaces of all routers in a virtual router must be in the same IP subnet.
- The IP address(es) 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.

Configuring parameters specific to VRRPE

VRRPE is configured at the interface level. To implement a simple VRRPE configuration using all the default values, enter commands such as the following on each device.

```
BigIron RX(config)# router vrrp-extended
BigIron RX(config)# inter e 1/5
BigIron RX(config-if-e10000-1/5)# ip address 192.53.5.3
BigIron RX(config-if-e10000-1/5)# ip vrrp-extended vrid 1
BigIron RX(config-if-e10000-1/5-vrid-1)# backup priority 50 track-priority 10
BigIron RX(config-if-e10000-1/5-vrid-1)# ip-address 192.53.5.254
BigIron RX(config-if-e10000-1/5-vrid-1)# activate
```

Syntax: ip vrrp-extended vrid <*vrid*>

Syntax: backup [priority <*value*>] [track-priority <*value*>]

Refer to "Authentication type" on page 451 for information on the **auth-type no-auth** | **simple-text-auth** <*auth-data*> parameters.

Also, refer to "Configuration rules for VRRPE" on page 450 additional information on how to configure VRRPE.

device requires you to identify a VRRPE 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.

Configuration rules for VRRPE

- 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 device devices.
- The Hello interval must be set to the same value on all the device devices.
- The Dead interval must be set to the same value on all the device devices.
- The track priority for a virtual router must be lower than the VRRPE priority.

NOTE
If you disable VRRPE, 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 after disabling the protocol, all configuration information for the disabled protocol is removed from the startup configuration.

Configuring additional VRRP and VRRPE parameters

You can modify the following VRRP and VRRPE 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

Refer to "VRRP and VRRPE parameters" on page 446 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 VRRPE packets on those interfaces also must use the same authentication. Brocade's implementation of VRRP and VRRPE supports the following authentication types:

- *No authentication* – The interfaces do not use authentication. This is the default for VRRP and VRRPE.
- *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)# inter e 1/6
Router1(config-if-e10000-1/6)# ip vrrp auth-type simple-text-auth ourpword
```

Configuring router 2

```
Router2(config)# inter e 1/5
Router2(config-if-e10000-1/5)# ip vrrp auth-type simple-text-auth ourpword
```

Syntax: 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.

Suppression of RIP advertisements on backup routers for the backup up interface

Normally, a VRRP or VRRPE 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: use-vrrp-path

The syntax is the same for VRRP and VRRPE.

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 should change the Dead interval on the Backup routers.

To change the Hello interval on the Master to 10 seconds, enter the following commands.

```
Router1(config)# inter e 1/6
Router1(config-if-e10000-1/6)# ip vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# hello-interval 10
```

Syntax: hello-interval <*value*>

The Hello interval can be from 1 – 84 seconds. The default is 1 second.

The syntax is the same for VRRP and VRRPE.

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. When Backups determine that the Master is dead, the Backup with the highest priority becomes the new Master. The Dead interval can be from 1 – 84 seconds. The default is 3.5 seconds. This 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, enter the following commands.

```
Router2(config)# inter e 1/5
Router2(config-if-e10000-1/5)# ip vrrp vrid 1
Router2(config-if-e10000-1/5-vrid-1)# dead-interval 30
```

Syntax: dead-interval <*value*>

The Dead interval can be from 1 – 84 seconds. The default is 3.5 seconds.

The syntax is the same for VRRP and VRRPE.

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 commands such as the following.

```
BigIron RX(config)# router vrrp
BigIron RX(config)# inter e 1/6
BigIron RX(config-if-e10000-1/6)# ip vrrp vrid 1
BigIron RX(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 commands such as the following.

```
BigIron RX(config)# router vrrp
BigIron RX(config)# inter e 1/6
BigIron RX(config-if-e10000-1/6)# ip vrrp vrid 1
BigIron RX(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 – 3600 seconds. The default is 60 seconds.

The syntax is the same for VRRP and VRRPE.

Track port

You can configure the virtual router to track the link state of interfaces on the device. 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 442.

To configure 1/6 on Router1 to track interface 2/4, enter the following commands.

```
Router1(config)# inter e 1/6
Router1(config-if-e10000-1/6)# ip vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# track-port e 2/4
```

Syntax: track-port ethernet <*slot*>/<*portnum*> ve <*num*>

The syntax is the same for VRRP and VRRPE.

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 VRRPE 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 VRRPE, the software reduces the virtual router priority by the amount of the priority of the tracked interface that went down. For example, if the VRRPE interface's priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VRRPE 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 453.

Syntax: owner [track-priority <*value*>]

Syntax: backup [priority <*value*>] [track-priority <*value*>]

The syntax is the same for VRRP and VRRPE.

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, 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.

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 commands such as the following.

```
Router1(config)# inter e 1/6
Router1(config-if-e10000-1/6)# ip vrrp vrid 1
Router1(config-if-e10000-1/6-vrid-1)# non-preempt-mode
```

Syntax: non-preempt-mode

The syntax is the same for VRRP and VRRPE.

Master router abdication and reinstatement

To change the Master's priority, enter commands such as the following.

```
BigIron RX(config)# ip int eth 1/6
BigIron RX(config-if-e10000-1/6)# ip vrrp vrid 1
BigIron RX(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 – 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.

```
BigIron RX(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 device is the Owner of the virtual router ("mode owner"), the device'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.

```
BigIron RX(config-if-e10000-1/6-vrid-1)# no owner priority 99
```

You cannot set the priority to 255 using the **owner priority** command.

Displaying VRRP and VRRPE information

You can display the following information for VRRP or VRRPE:

- Summary configuration and status information
- Detailed configuration and status information
- VRRP and VRRPE Statistics

Displaying summary information

To display summary information for a device, enter the following command at any level of the CLI.

```
BigIron RX(config)# show ip vrrp-extended brief
Total number of VRRP-Extended routers defined: 41

Inte- VRID Current   P  State   Master IP     Backup IP   Virtual IP
rface      Priority             Address       Address     Address
--------------------------------------------------------------------------------
v21   21   95        P  Backup  172.16.51.2   Local      172.16.51.1
v22   22   95        P  Backup  172.16.52.2   Local      172.16.52.1
v23   23   95        P  Backup  172.16.53.2   Local      172.16.53.1
v24   24   95        P  Backup  172.16.54.2   Local      172.16.54.1
v25   25   95        P  Backup  172.16.55.2   Local      172.16.55.1
v26   26   95        P  Backup  172.16.56.2   Local      172.16.56.1
v27   27   95        P  Backup  172.16.57.2   Local      172.16.57.1
```

Syntax: show ip vrrp [brief | ethernet <*slot*>/<*portnum*> | ve <*num*> | stat]

Syntax: show ip vrrp-extended [brief | ethernet <*slot*>/<*portnum*> | ve <*num*> | stat]

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 457.

The **ethernet** <*slot*>/<*portnum*> parameter specifies an Ethernet port. If you use this parameter, the command displays VRRP or VRRPE information only for the specified port.

The **ve** <*num*> parameter specifies a virtual interface. If you use this parameter, the command displays VRRP or VRRPE information only for the specified virtual interface.

The **stat** parameter displays statistics. Refer to "Displaying statistics" on page 460.

This display shows the following information.

TABLE 85 CLI display of VRRP or VRRPE summary information

This field...	Displays...
Total number of VRRP (or VRRP-Extended) routers defined	The total number of virtual routers configured on this device. **NOTE:** The total applies only to the protocol the device is running. For example, if the device is running VRRPE, the total applies only to VRRPE routers.
Interface	The interface on which VRRP or VRRPE is configured. If VRRP or VRRPE 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.

TABLE 85 CLI display of VRRP or VRRPE summary information (Continued)

This field...	Displays...
CurPri	The current VRRP or VRRPE priority of this device for the virtual router.
P	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.
State	This device's VRRP or VRRPE 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 device is a Backup for the virtual router. • **Master** – This device 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, enter the following command at any level of the CLI.

```
BigIron RX(config)# show ip vrrp-extended

Total number of VRRP-Extended routers defined: 41

Interface v21
---------------
auth-type no authentication

VRID 21 (index 1)
interface v21
state backup
administrative-status enabled
mode non-owner(backup)
virtual mac 02e0.520b.3515
priority 95
current priority 95
track-priority 24
hello-interval 1 sec
backup hello-interval 60 sec
advertise backup enabled
dead-interval 0 sec
current dead-interval 3.6 sec
preempt-mode true
virtual ip address 172.16.51.1
next backup hello sent in 6.5 sec
master router 172.16.51.2 expires in 3.3 sec
track-port 8/1(up) 4/1(up) 8/13(up) 16/1(up)
```

Syntax: show ip vrrp [brief | ethernet <*slot*>/<*portnum*> | ve <*num*> | stat]

Syntax: show ip vrrp-extended [brief | ethernet <*slot*>/<*portnum*> | ve <*num*> | stat]

The **brief** parameter displays summary information. Refer to "Displaying summary information" on page 456.

The **ethernet** <*slot*>/<*portnum*> parameter specifies an Ethernet port. If you use this parameter, the command displays VRRP or VRRPE information only for the specified port.

The **ve** <*num*> parameter specifies a virtual interface. If you use this parameter, the command displays VRRP or VRRPE information only for the specified virtual interface.

The **statistic** parameter displays statistics. Refer to "Displaying statistics" on page 460.

This display shows the following information.

TABLE 86 CLI display of VRRP or VRRPE detailed information

This field...	Displays...
Total number of VRRP (or VRRP-Extended) routers defined	The total number of virtual routers configured on this device. **NOTE:** The total applies only to the protocol the device is running. For example, if the device is running VRRPE, the total applies only to VRRPE routers.
Interface parameters	
Interface	The interface on which VRRP or VRRPE is configured. If VRRP or VRRPE 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.
state	This device's VRRP or VRRPE 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 device is a Backup for the virtual router. • **master** – This device 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 VRRPE has not been activated on the interface. • **enabled** – VRRP or VRRPE has been activated on the interface.

TABLE 86 CLI display of VRRP or VRRPE detailed information (Continued)

This field...	Displays...
mode	Indicates whether the device 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 device devices configured for VRRPE 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 VRRPE priority of this device for the virtual router. The current priority can differ from the configured priority (see the row above) 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 442.
track priority	VRRPE 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.
advertise backup	The IP addresses of Backups that have advertised themselves to this device 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 452.
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. **NOTE:** 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.

TABLE 86 CLI display of VRRP or VRRPE detailed information (Continued)

This field...	Displays...
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 device 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 device itself will become the Master. **NOTE:** This field applies only when this device is a Backup.
track port	The interfaces that the virtual router's interface is tracking. If the link for a tracked interface goes down, the VRRP or VRRPE 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.

Displaying statistics

To display VRRP statistics, enter the following command.

```
device#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 [brief | ethernet <*slot*>/<*portnum*> | ve <*num*> | stat]

Syntax: show ip vrrp-extended [brief | ethernet <*slot*>/<*portnum*> | ve <num> | stat]

The **brief** parameter displays the summary information. If you do not use this parameter, detailed information is displayed instead.

The **ethernet** <*slot*>/<*portnum*> parameter specifies an Ethernet port. If you use this parameter, the command displays VRRP information only for the specified port.

The **ve** <*num*> parameter specifies a virtual interface. If you use this parameter, the command displays VRRP information only for the specified virtual interface.

The **statistics** parameter displays statistics.

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

Clearing VRRP or VRRPE statistics

To clear VRRP or VRRPE statistics, enter the following command at the Privileged EXEC level or any configuration level of the CLI.

```
device(config)# clear ip vrrp
```

Syntax: clear ip vrrp-stat

Configuration examples

The following sections contain the CLI commands options for implementing the VRRP and VRRPE configurations shown in Figure 84 on page 441 and Figure 85 on page 445.

VRRP example

To implement the VRRP configuration shown in Figure 84 on page 441, enter the following commands.

Configuring Router1

To configure VRRP Router1, enter the following commands.

```
Router1(config)# router vrrp
Router1(config)# inter e 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. Because 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 84 on page 441 after enabling VRRP, enter the following commands.

```
Router2(config)# router vrrp
Router2(config)# inter e 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 442.

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

VRRPE example

To implement the VRRPE configuration shown in Figure 85 on page 445, configure the VRRP Routers as shown in the following sections.

Configuring Router1

To configure VRRP Router1 in Figure 85 on page 445, 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.

```
Router1(config)# router vrrp-extended
Router1(config)# interface ethernet 5/1
Router1(config-if-e10000-5/1)# ip address 192.53.5.3/24
Router1(config-if-e10000-5/1)# ip vrrp-extended vrid 1
Router1(config-if-e10000-5/1-vrid-1)# backup priority 100 track-priority 20
```

```
Router1(config-if-e10000-5/1-vrid-1)# track-port ethernet 3/2
Router1(config-if-e10000-5/1-vrid-1)# ip-address 192.53.5.254
Router1(config-if-e10000-5/1-vrid-1)# activate
Router1(config-if-e10000-5/1-vrid-1)# exit
Router1(config)# interface ethernet 5/1
Router1(config-if-e10000-5/1)# ip vrrp-extended vrid 2
Router1(config-if-e10000-5/1-vrid-1)# backup priority 110 track-priority 20
Router1(config-if-e10000-5/1-vrid-1)# track-port ethernet 2/4
Router1(config-if-e10000-5/1-vrid-1)# ip-address 192.53.5.253
Router1(config-if-e10000-5/1-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRPE 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 VRRPE priority in relation to the other VRRPE routers in this virtual router. The **track-priority** parameter specifies the new VRRPE priority that the router receives for this virtual router if the interface goes down. Refer to "Track ports and track priority" on page 442.

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 VRRPE configuration. Alternatively, you can use the **enable** command. The **activate** and **enable** commands do the same thing.

Syntax: router vrrp-extended

Syntax: ip vrrp-extended vrid <vrid>

Syntax: backup [priority <value>] [track-priority <value>]

Syntax: track-port ethernet <slot>/<portnum> ve <num>

Syntax: ip-address <ip-addr>

Syntax: activate

Chapter

Configuring Quality of Service 18

In this chapter

Overview of Quality of Service (QoS)

Quality of Service (QoS) features are used to prioritize the use of bandwidth in a switch. When QoS features are enabled, traffic is classified as it arrives at the switch, and processed through on the basis of configured priorities. Traffic can be dropped, prioritized for guaranteed delivery, or subject to limited delivery options as configured by a number of different mechanisms.

Classification

Classification is the process of selecting packets on which to perform QoS, reading the QoS information and assigning them a priority. The classification process assigns a priority to packets as they enter the switch. These priorities can be determined on the basis of information contained within the packet or assigned to the packet as it arrives at the switch. Once a packet or traffic flow is classified, it is mapped to one of four forwarding priority queues.

Packets on the BigIron RX are classified in up to eight traffic classes with values between 0 and 7. Packets with higher priority classifications are given a precedence for forwarding. These classes are determined by the following criteria in ascending order:

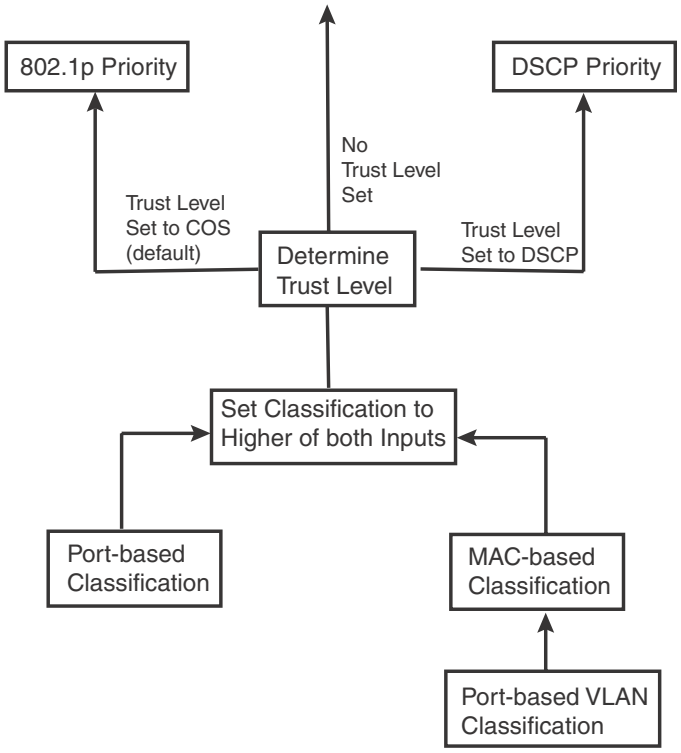
- **Configured port priority** – A priority can be set for all traffic that arrives at a port. This is implemented through the interface configuration.
- **VLAN priority** – A priority can be set for a specified port-based VLAN in the VLAN configuration.

- **Packet Source MAC address** – A priority can be set for a specified MAC address by assigning a static MAC entry to a specific priority in the VLAN configuration. Note: This priority affects packets sourced by this MAC address and not packed destined for this MAC address.
- **Packet priority** – Depending on the Trust level set, a packet can be classified by either the 802.1p priority or DSCP value that it has when it arrives at the switch. If no trust level is set, the packet will default to a priority set by earlier criteria. By default, the trust level is set to 802.1p. In addition, you can configure a port to override the DSCP value for every packet that arrives on it to a user-configured value.

Processing of classified traffic

Given the variety of different criteria, there are multiple possibilities for traffic classification within a stream of network traffic. For this reason, the priority of packets must be resolved based on which criteria takes precedence. Precedence follows the scheme illustrated in Figure 86.

FIGURE 86 Priority resolution



As shown in the figure, the first criteria considered are port-based, MAC-based, and port-based VLAN classifications. The packet is primarily classified with the higher of these two criteria. Next, the packet is classified based on the trust level set. If there is no trust level set, the packet retains the port or MAC derived QoS classification, If a trust level is set, the packet will either take the 802.1p or DSCP priority depending on which is set as the trust level.

Once a packet is classified by one of the procedures mentioned, it is mapped to an internal forwarding queue in the device. There are four queues designated as 0 to 3. The internal forwarding priority maps to one of these four queues as shown in Table 87 through Table 90. The mapping between the internal priority and the forwarding queue cannot be changed.

Table 87 through Table 90 show the default QoS mappings on the device, which are used if the trust level for CoS or DSCP is enabled.

TABLE 87 Default QoS mappings, columns 0 to 15

DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15
802.1p (COS) Value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15
Internal Forwarding Priority	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Forwarding Queue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE 88 Default QoS mappings, columns 16 to 31

DSCP value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
802.1p (COS) Value	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
DSCP value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Internal Forwarding Priority	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
Forwarding Queue	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLE 89 Default QoS mappings, columns 32 to 47

DSCP value	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
802.1p (COS) Value	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
DSCP value	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Internal Forwarding Priority	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
Forwarding Queue	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

TABLE 90 Default QoS mappings, columns 48 to 63

DSCP value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
802.1p (COS) Value	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
DSCP value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Internal Forwarding Priority	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
Forwarding Queue	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

The mapping between the internal forwarding priority and values received and forwarded can be changed as follows:

- **COS to DSCP Mapping** – You can change the mapping between 802.1p (COS) values from the default values shown in Table 87 through Table 90. This mapping is used for DSCP marking when trust level is COS. Refer to "Changing the CoS –> DSCP mappings" on page 471.
- **DSCP to DSCP Mapping** – You can alter the DSCP value of a packet that is received to a value configured on the switch. This mapping is used for DSCP marking when trust level is DSCP. Refer to "Changing the DSCP –> DSCP mappings" on page 472.
- **DSCP to Internal Forwarding Priority Mapping** – You can change the mapping between the DSCP value and the Internal Forwarding priority value from the default values shown in Table 87 through Table 90. This mapping is used for COS marking and determining the internal priority when the trust level is DSCP. Refer to "Changing the DSCP –> internal forwarding priority mappings" on page 472.
- **COS to Internal Forwarding Priority Mapping** – You can change the mapping between 802.1p (COS) values and the Internal Forwarding priority value from the default values shown in Table 87 through Table 90. This mapping is used for COS marking and determining the internal priority when the trust level is COS. "Changing the CoS –> internal forwarding priority mappings" on page 473.

Marking

Marking is the process of changing the packet's QoS information (the 802.1p and DSCP information in a packet) for the next hop. You can mark a packet's Layer 2 CoS value, its Layer 3 DSCP value, or both values. The Layer 2 CoS or DSCP value the device marks in the packet is the same value that results from mapping the packet's QoS value into a Layer 2 CoS or DSCP value.

Marking is optional and is disabled by default. When marking is disabled, the device still performs mappings for scheduling the packet, but leaves the packet's QoS values unchanged when the device forwards the packet.

Configuring DSCP classification by interface

You can configure DSCP classification on an interface to set the DSCP value of every packet that arrives on the interface to a value that you configure. After the packet's DSCP value has been set using this command, it is subject to classification, marking, and scheduling operations that are configured.

To configure the 1/1 interface to set all packets that arrive on it to a DSCP value of 23, use the following command.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# dscp 23
```

Syntax: [no] dscp <*num*>

The <*num*> parameter can be any possible DSCP value from 0 to 63.

Configuring port, MAC, and VLAN-based classification

Assigning QoS priorities to traffic

By default, traffic is forwarded using the best-effort queue (qosp0). However, traffic can be classified into different priorities, based on the following:

- Incoming port (sometimes called the ingress port)
- Port-based VLAN membership
- Static MAC entry

The following sections describe how to change the priority for each of the items listed above.

Although it is possible for a packet to qualify for an adjusted QoS priority based on more than one of the criteria above, the system determines the priority it will use for forwarding as described in "Processing of classified traffic" on page 466.

When you apply a QoS priority to one of the items listed above, you specify a number from 0 – 7. The priority number specifies the IEEE 802.1p equivalent to one of the four Brocade QoS queues. The numbers correspond to the queues as follows.

Priority level	QoS forwarding queue
6, 7	3
4, 5	2
2, 3	1
0, 1	0

Changing a port's priority

To change a port's QoS priority, use one of the following methods. The priority applies to inbound traffic on the port. The default priority of each port is 0.

To change the QoS priority of port 1/1 on a device to queue 2, enter the following commands.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# priority 5
```

Syntax: [no] priority <*num*>

The <*num*> parameter can be from 0 – 7 and specifies the priority level equivalent to one of the four QoS queues.

Changing a Layer 2 port-based VLAN's priority

By default, VLANs have priority 0. To change a port-based VLAN's QoS priority, use one of the following methods. The priority applies to inbound traffic on ports in the VLAN.

To change the QoS priority of port-based VLAN 20 to queue 3, enter the following commands.

```
BigIron RX(config)# vlan 20
BigIron RX(config-vlan-20)# priority 7
```

Syntax: [no] priority <*num*>

The <*num*> parameter can be from 0 – 7 and specifies the priority level equivalent to one of the four QoS queues.

Assigning static MAC address entries to priority queues

By default, all MAC address entries are in the best effort queue. When you configure a static MAC entry, you can assign the entry to a higher QoS level using the following method. The priority applies to packets sourced by this MAC address.

To configure a static MAC entry and assign the entry to the premium queue on a Chassis device, enter commands such as the following.

```
BigIron RX(config)# vlan 9
BigIron RX(config-vlan-9)# static-mac-address 1145.1163.67FF ethernet 1/1
priority 7
```

Syntax: [no] static-mac-address <*mac-addr*> ethernet <*slot*>/<*portnum*> [priority <*num*>] [host-type | router-type | fixed-host]

The <*num*> parameter can be from 0 – 7 and specifies the priority level1 equivalent to one of the four QoS queues.

Configuring ToS-based QoS

To configure ToS-based QoS, perform the following tasks:

- Enable ToS-based QoS on an interface. Once you enable the feature on an individual interface, you can configure the trust level and marking for traffic that is received on that interface as described:
  - Specify the trust level for packets received on the interface.
  - Enable marking of packets received on the interface.

Enabling ToS-based QoS

To enable ToS-based QoS on an interface, enter the following command at the configuration level for the interface.

```
BigIron RX(config-if-e1000-1/1)# qos-tos
```

Syntax: [no] qos-tos

Specifying trust level

If a packet arrives on the interface with either a COS, DSCP, or COS and DSCP priority level, the trust level specifies which of these priorities you want to accept. If you disable trust level, the priority will default to a criteria other than the COS or DSCP priority.

To set the trust level for an interface to dscp, enter the following command at the configuration level for the interface.

```
BigIron RX(config-if-e1000-1/1)# qos-tos trust dscp
```

Syntax: [no] qos-tos trust cos | dscp

The **cos | dscp** parameter specifies the trust level:

- **cos** – The device uses the 802.1p (CoS) priority value in the packet's Ethernet frame header to determine the packet's internal forwarding priority. This is the default state and is in effect even Qos-ToS is enabled on a port.
- **dscp** – The device uses the six most-significant bits in the packet's ToS field and interprets them as a DSCP value to determine the packet's internal forwarding priority.

Enabling marking

This command enables marking of the 802.1p field or the DSCP field in the ToS byte of an IP header.

Syntax: [no] qos-tos mark cos | dscp

The **cos | dscp** parameter specifies the type of marking:

- **cos** – The device changes the outbound packet's 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 DSCP value to match the results of the device's QoS mapping from the specified trust level.

Configuring the QoS mappings

The Brocade device maps a packet's 802.1p or DSCP value to an internal forwarding priority. The default mappings are listed in Table 87 through Table 90. You can change the following mappings as described in this section:

- CoS –> DSCP
- DSCP –> DSCP
- DSCP –> internal forwarding priority
- CoS –> internal forwarding priority

The mappings are globally configurable and apply to all interfaces.

NOTE
In a configuration where you have marking enabled with the trust level set to CoS, 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.

NOTE
The mappings are globally configurable and apply to all interfaces.

Changing the CoS –> DSCP mappings

The CoS –> DSCP mappings are used if the trust level is CoS and DSCP marking is enabled.

To change the CoS –> DSCP mappings, enter commands such as the following at the global CONFIG level of the CLI.

```
BigIron RX(config)# qos-tos map cos-dscp 0 33 25 49 17 7 55 41
BigIron RX(config)# ip rebind-acl all
```

This command configures the mappings displayed in the COS-DSCP map portion of the QoS information display.

```
BigIron RX(config-if-e10000-1/1)# 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 <dscp0> <dscp1> <dscp2> <dscp3> <dscp4> <dscp5> <dscp6> <dscp7>

The <dscp0> through <dscp7> parameters specify the DSCP values you are mapping the eight CoS values to. You must enter DSCP values for all eight CoS values, in order from CoS value 0 – 7.

Changing the DSCP –> DSCP mappings

The DSCP –> DSCP mappings are used when DSCP trust level and DSCP marking are enabled. To change a DSCP –> DSCP mapping, enter a command such as the following at the global CONFIG level of the CLI.

```
BigIron RX(config)# qos-tos map dscp-dscp 0 to 10
```

This command changes the mapping of DSCP value 0 to 10.

Syntax: [no] qos-tos map dscp-dscp <old-dscp-value> [<old-dscp-value>...] to <new-dscp-value>

You can change up to seven DSCP values in the same commend.

Changing the DSCP –> internal forwarding priority mappings

This mapping is used when the trust level is set to DSCP. In addition to determining the internal-forwarding priority of a packet, the value also determines the outbound 802.1p value if CoS marking is enabled. To change the DSCP –> internal forwarding priority mappings for all the DSCP ranges, enter commands such as the following at the global CONFIG level of the CLI.

```
BigIron RX(config)# qos-tos map dscp-priority 0 2 3 4 to 1
BigIron RX(config)# qos-tos map dscp-priority 8 to 5
BigIron RX(config)# qos-tos map dscp-priority 16 to 4
BigIron RX(config)# qos-tos map dscp-priority 24 to 2
BigIron RX(config)# qos-tos map dscp-priority 32 to 0
BigIron RX(config)# qos-tos map dscp-priority 40 to 7
BigIron RX(config)# qos-tos map dscp-priority 48 to 3
BigIron RX(config)# qos-tos map dscp-priority 56 to 6
```

These commands configure the mappings displayed in the DSCP to forwarding priority portion of the QoS information display. To read this part of the display, select the first part of the DSCP value from the d1 column and select the second part of the DSCP value from the d2 row. For example, to read the DSCP to forwarding priority mapping for DSCP value 24, select 2 from the d1 column and select 4 from the d2 row. The mappings that are changed by the command above are shown below in bold type.

```
BigIron RX(config-if-e10000-1/1)# show qos-tos
```

...portions of table omitted for simplicity...

```
DSCP-Priority map: (dscp = d1d2)

     d2|  0   1   2   3   4   5   6   7   8   9
  d1   |
  -----+----------------------------------------
   0   |  1   0   1   1   1   0   0   0   5   1
   1   |  6   1   1   1   1   1   4   2   2   2
   2   |  2   2   2   2   2   3   3   3   3   3
   3   |  3   3   0   4   4   4   4   4   4   4
   4   |  7   5   5   5   5   5   5   5   3   6
   5   |  6   6   6   6   6   6   6   7   7   7
   6   |  7   7   7   7
```

For information about the rest of this display, refer to "Displaying QoS configuration information" on page 474.

Syntax: [no] qos-tos map dscp-priority <dscp-value> [<dscp-value>...] to <priority>

The <dscp-value> [<dscp-value>...] parameter specifies the DSCP value ranges you are remapping. You can specify up to seven DSCP values in the same command, to map to the same forwarding priority. The first command in the example above maps priority 1 to DSCP values 0, 2, 3, and 4.

The <priority> parameter specifies the internal forwarding priority.

Changing the CoS –> internal forwarding priority mappings

This mapping is used when the trust level is set to CoS. In addition to determining the internal-forwarding priority of a packet, the value also determines the outbound 802.1p value if CoS marking is enabled. To change the
CoS –> internal forwarding priority mappings for all the CoS ranges, enter commands such as the following at the global CONFIG level of the CLI.

```
BigIron RX(config)# qos-tos map cos-priority 7 4 3 6 5 2 1 0
```

These commands configure the mappings displayed in the CoS to forwarding priority portion of the QoS information display. To read this part of the display, select the first part of the CoS value from the d1 column and select the second part of the CoS value from the d2 row. For example, to read the CoS to forwarding priority mapping for CoS value 24, select 2 from the d1 column and select 4 from the d2 row. The mappings that are changed by the command above are shown below in bold type.

```
BigIron RX(config-if-e10000-1/1)# show qos-tos
```

...portions of table omitted for simplicity...

```
COS-Priority map:
COS:      0 1 2 3 4 5 6 7
-----------------------------------------------
Priority: 0 1 2 3 4 5 6 7
```

For information about the rest of this display, refer to "Displaying QoS configuration information" on page 474.

Syntax: [no] qos-tos map cos-priority <prio0> <prio1><prio2><prio3><prio4><prio5><prio6> <prio7>

The <prio0> through <prio7> parameters specify the COS values you are mapping the eight internal priorities to. You must enter CoS values for all eight internal priorities, in order from priority 0 – 7.

Displaying QoS configuration information

To display configuration information, enter the following command at any level of the CLI.

```
BigIron RX# show qos-tos
Interface QoS , Marking and Trust Level:

  i/f  | QoS |   Mark   |  Trust-Level
-------+-----+----------+---------------
1/2    | Yes |          |  Layer 2 CoS
ve1    | No  |          |  Layer 2 CoS
ve4    | No  |          |  Layer 2 CoS
ve5    | No  |          |  Layer 2 CoS
ve20   | No  |          |  Layer 2 CoS
COS-DSCP map:
COS:  0  1  2  3  4  5  6  7
-----------------------------------------------
dscp: 0  8 16 24 32 40 48 56

DSCP-Priority map: (dscp = d1d2)
d2 | 0  1  2  3  4  5  6  7  8  9
d1 |
-----+---------------------------------------
0  | 0  0  0  0  0  0  0  0  1  1
1  | 1  1  1  1  1  1  2  2  2  2
2  | 2  2  2  2  3  3  3  3  3  3
3  | 3  3  4  4  4  4  4  4  4  4
4  | 5  5  5  5  5  5  5  5  6  6
5  | 6  6  6  6  6  6  7  7  7  7
6  | 7  7  7  7
DSCP-DSCP map: (dscp = d1d2)
d2 | 0  1  2  3  4 5 6 7 8 9
d1 |
-----+---------------------------------------
0  | 0  1  2  3  4  5  6  7  8  9
1  | 10 11 12 13 14 15 16 17 18 19
2  | 20 21 22 23 24 25 26 27 28 29
3  | 30 31 32 33 34 35 36 37 38 39
4  | 40 41 42 43 44 45 46 47 48 49
5  | 50 51 52 53 54 55 56 57 58 59
6  | 60 61 62 63
COS-Priority map:
COS:      0 1 2 3 4 5 6 7
-----------------------------------------------
Priority: 0 1 2 3 4 5 6 7
```

Syntax: show qos-tos

This command shows the following information.

TABLE 91 ToS-based QoS configuration information

This field...	Displays...
Interface QoS, marking and trust level information	
i/f	The interface
QoS	The state of ToS-based QoS on the interface. The state can be one of the following: • **No** – Disabled • **Yes** – Enabled
Mark	The marking type enabled on the interface. The marking type can be any of the following: • **COS** – CoS marking is enabled. • **DSCP** – DSCP marking is enabled. • **No** – Marking is not enabled.
Trust-Level	The trust level enabled on the interface. The trust level can be one of the following: • DSCP • L2 CoS
CoS-DSCP map	
COS	The CoS (802.1p) values.
dscp	The DSCP values to which the device maps the CoS values above.
DSCP-priority map	
d1 and d2	The DSCP -> forwarding priority mappings that are currently in effect.
DSCP-DSCP map	
d1 and d2	The DSCP -> DSCP mappings that are currently in effect.
CoS-priority map	The CoS (802.1p) forwarding priority mapping that is currently in effect.

Determining packet drop priority using WRED

You can configure a `device` to monitor traffic congestion and drop packets according to a WRED (Weighted Random Early Detection) algorithm. This algorithm enables the system to detect the onset of congestion and take corrective action. In practice, WRED causes a Switch to start dropping packets as traffic in the switch 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:

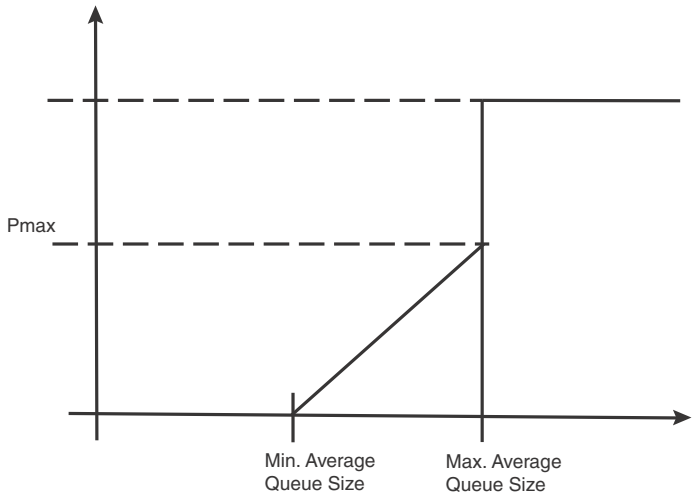
- **Statistical Average-Q-Size** – The statistical average size of the queue calculated over time on the switch.
- **Current-Q-Size** – The current size of the queue as calculated on the switch.
- **Wq** – This variable specifies the weights that should be given to the current queue size and the statistical 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 WRED Operates

The graph in Figure 87 describes the interaction of the previously described variables in the operation of WRED. When a packet arrives at a switch, 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 476). 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 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 477.

FIGURE 87 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 **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 reduce 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 switch.

$$Pdrop = \frac{pkt\text{-}size}{pkt\text{-}size\text{-}max} * Pmax * \frac{(avg\text{-}q\text{-}size - min\text{-}avg\text{-}q\ size)}{(max\text{-}avg\text{-}q\text{-}size - min\text{-}avg\text{-}q\ size)}$$

Using WRED with rate limiting

When rate limiting is configured on a device, it directs the switch to drop traffic indiscriminately when the configured **average-rate** and **maximum-burst** thresholds are exceeded. If rate limiting is configured with WRED, the traffic that exceeds these thresholds can be subjected to the WRED algorithm which drops packets selectively by priority.

In this configuration, packets that exceed the thresholds established by the rate limiting configuration are marked as either exceeding the **average-rate** or **maximum-burst** threshold. This marking is then used to select a WRED configuration that determines which packets to drop.

Configuring packet drop priority using WRED

For a description of WRED, refer to "Determining packet drop priority using WRED" on page 475. This section describes how to configure the parameters described in that section to enable the use of WRED on a `device`.

To configure WRED, you must configure the following parameters:

- "Enabling WRED"
- "Setting the averaging-weight (Wq) parameter"
- "Configuring the drop precedence parameters"

Enabling WRED

WRED must be enabled on any forwarding queue that you want it to operate on. To enable WRED for the forwarding queue 3, enter the following command.

```
BigIron RX(config)#qos queue-type 3 wred enable
```

Syntax: [no] qos queue-type <queue-number> wred enable

The <queue-number> variable is the number of the forwarding queue that you want to enable WRED for. There are four forwarding queues on device. They are numbered 0 to 3 with zero as the lowest priority queue and three the highest.

Setting the averaging-weight (Wq) parameter

The Wq parameter 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 92

TABLE 92 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.

```
BigIron RX(config)#qos queue-type 1 wred averaging-weight 25%
```

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 device Routers. They are numbered 0 to 3.

The <*avg-weight-value*> variable is the weight-ratio between instantaneous and average queue sizes. It can be one of the 13 values expressed as 1 to 13 described in Table 92. The default value is 9 which maps to a Wq value of.19%.

Configuring the drop precedence parameters

The DSCP/TOS bits in packets are used to prioritize packet delivery for specified queue types. These values are from 0 to 3. Packets with a DSCP/TOS value of 0 are least likely to be dropped and packets with a DSCP/TOS of 3 are most likely to be dropped.

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/TOS value. The following sections describe how to set the following drop precedence parameters for each of the four DSCP/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"
- Packets that do not have the DSCP/TOS value set are assigned a drop precedence equal to the DSCP/TOS level of 0.

Setting the maximum drop probability

To set the maximum drop probability when the queue size reaches the Max-average-q-size value to 20% use the following command.

```
BigIron RX(config)#qos queue-type 1 wred drop-precedence 0 drop-probability-max
20%
```

Syntax: [no] qos queue-type <queue-number> wred drop-precedence <policing-status> drop-probability-max <p-max%>

The <*queue-number*> variable is the number of the forwarding queue that you want to configure drop-precedence for. There are four forwarding queues on device. They are numbered 0 to 3 with zero as the lowest priority queue and three the highest.

The <*policing-status*> variable indicates the traffic policing status for which you want to configure drop-precedence.

The <*p-max%*> variable defines the maximum drop probability in when the queue size is at the value configured for max-avg-q-size. This value is expressed as a percentage.

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.

```
BigIron RX(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 device Routers. They are numbered 0 to 3.

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 93.

Setting the minimum and maximum average queue size

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.

```
BigIron RX(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.

```
BigIron RX(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 device Routers. They are numbered 0 to 3.

The <*drop-precedence-value*> variable for the **drop-precedence** parameter is the TOS/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 93.

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 93.

Setting the maximum packet size

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.

```
BigIron RX(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 BigIron RX Routers. They are numbered 0 to 3.

The <*drop-precedence-value*> variable for the drop-precedence parameter is the TOS/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 93.

Restoring to default WRED parameters

Table 93 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.

```
BigIron RX(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 four forwarding queues on device Routers. They are numbered 0 to 3.

TABLE 93 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	Average weight
0	0	356	1024	16384	2%	1024	0.2%
	1	304	1024	16384	4%		
	2	256	1024	16384	9%		
	3	204	1024	16384	10%		
1	0	356	1024	16384	2%	1024	0.2%
	1	304	1024	16384	4%		
	2	256	1024	16384	9%		
	3	204	1024	16384	10%		
2	0	408	1024	16384	2%	1024	0.2%
	1	356	1024	16384	4%		
	2	304	1024	16384	9%		
	3	256	1024	16384	9%		
3	0	408	1024	16384	2%	1024	0.2%
	1	356	1024	16384	4%		
	2	304	1024	16384	9%		
	3	256	1024	16384	9%		

Displaying the WRED configuration

To view a WRED configuration, use the following command.

```
BigIron RX#show qos  wred
QType Enable AverWt MaxQSz DropPrec MinAvgQSz MaxAvgQSz MaxDropProb MaxPktSz
0     Yes    100%   32768   0       16000     32000      30%         512
                             1       32768     32768       0%         512
                             2       32768     32768     100%         512
                             3       32768     32768       0%         512
1     Yes    100%   32768   0       32768     32768       0%         512
                             1       32768     32768       0%         512
                             2       24000     24000      50%         512
                             3       32768     32768       0%         512
2     No
3     No
```

Syntax: show qos wred

Scheduling traffic for forwarding

If the traffic being processed by a device is within the capacity of the switch, all traffic is forwarded as received. Once we reach the point where the switch is bandwidth constrained, it becomes subject to drop priority if configured as described in "Determining packet drop priority using WRED" on page 475 or traffic scheduling as described in this section.

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.
- **Enhanced strict scheduling** – With enhanced strict scheduling enabled, a configurable minimum bandwidth is allocated to lower-priority traffic so that it is not starved. The remaining bandwidth is used in a strict scheduling manner.
- **WFQ destination-based 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.
- **WFQ source-based scheduling** – With WFQ source-based scheduling enabled, some weight-based bandwidth is allocated to all queues. With this scheme, the configured weight distribution from an input port is guaranteed allocation in relationship to the configured weight distribution. However, because multiple input ports can aggregate traffic to a single output port, the traffic egressing a single port may not equal the configured values.
- **Maximum rate-based scheduling** – With maximum rate-based scheduling enabled, a configured maximum bandwidth is allocated to each priority level. Bandwidth remaining after the aggregate maximum is allocated is **not used.**
- **Minimum rate-based scheduling** – With minimum rate-based scheduling enabled, a configured minimum bandwidth is allocated to each priority level. Bandwidth remaining after the aggregate minimum is allocated is redistributed equally among the four priority queues.

Configuring traffic scheduling

Traffic scheduling is configured on a per-port basis. The following sections describe how to configure each of the traffic scheduling schemes:

- "Configuring strict priority-based traffic scheduling"
- "Configuring enhanced strict priority-based traffic scheduling"
- "Calculating the values for WFQ source and destination-based traffic scheduling"
- "Configuring WFQ destination-based traffic scheduling"
- "Configuring WFQ source-based traffic scheduling"
- "Configuring maximum rate-based traffic scheduling"
- "Configuring minimum rate-based traffic scheduling"

NOTE
Brocade only support "strict" and "destination-weighted" scheduling schemes. (qos scheduler ..) on the 16 x 10G modules.

Configuring strict priority-based traffic scheduling

To configure strict priority-based scheduling use a command such as the following.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# qos scheduler strict
```

Syntax: qos scheduler strict

Configuring enhanced strict priority-based traffic scheduling

To configure enhanced strict priority-based scheduling use a command such as the following.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# qos scheduler enhanced-strict 100 100 100
```

Syntax: qos scheduler enhanced-strict <Queue0-rate> <Queue1-rate> <Queue2-rate>

The <*Queue0-rate*> variable defines the minimum bandwidth allocated to lower-priority traffic rate in Kbps for forwarding queue 0.

The <*Queue1-rate*> variable defines the minimum bandwidth allocated to lower-priority traffic rate in Kbps for forwarding queue 1.

The <*Queue2-rate*> variable defines the minimum bandwidth allocated to lower-priority traffic rate in Kbps for forwarding queue 2.

Calculating the values for WFQ source and destination-based traffic scheduling

Weighted Fair Queueing (WFQ) scheduling is configured to be a percentage of available bandwidth using the following formula.

$$\text{Weight of } q(x) = \frac{q(x)}{q0 + q1 + q2 + q3}$$

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 - 3).

q0 - q3 = the assigned values of the four 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 3.

- Queue 0 = 5, Queue 1 = 10, Queue 2 = 15, and Queue 3 = 20

NOTE
Where rates are configured, the minimum rate supported is 248 Kbps for 1 Gbps ports and 2480 Kbps for 10 Gbps ports.

To determine the weight of **q3**.

$$\text{Weight of q3} = \frac{20}{5 + 10 + 15 + 20}$$

The weight of q3 is 40%. Consequently, q3 will get 40% of the port's total bandwidth.

The values of the remaining queues are calculated to be the following.

q2 = 30%, q1 = 20%, and q0 = 10%

Configuring WFQ destination-based traffic scheduling

To configure WFQ destination-based scheduling use a command such as the following.

```
BigIron RX(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# qos scheduler destination-weighted 5 10 15 20
```

Syntax: qos scheduler destination-weighted <queue0-weight> <queue1-weight> <queue2-weight> <queue3-weight>

The <*queue0-weight*> variable defines the relative value for queue0 in calculating queue0's allocated bandwidth.

The <*queue1-weight*> variable defines the relative value for queue1 in calculating queue1's allocated bandwidth.

The <*queue2-weight*> variable defines the relative value for queue2 in calculating queue2's allocated bandwidth.

The <*queue3-weight*> variable defines the relative value for queue3 in calculating queue3's allocated bandwidth.

Refer to "Calculating the values for WFQ source and destination-based traffic scheduling" for information on assigning *queue0-weight* to *queue3-weight* values.

Configuring WFQ source-based traffic scheduling

To configure WFQ source-based scheduling use a command such as the following.

```
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# qos scheduler source-weighted 25 25 25 25
```

Syntax: qos scheduler source-weighted <Queue0-weight> <Queue1-weight> <Queue2-weight> <Queue3-weight>

The <Queue0-weight> variable defines the relative value for queue0 in calculating queue0's allocated bandwidth.

The <Queue1-weight> variable defines the relative value for queue1 in calculating queue1's allocated bandwidth.

The <Queue2-weight> variable defines the relative value for queue2 in calculating queue2's allocated bandwidth.

The <Queue3-weight> variable defines the relative value for queue3 in calculating queue3's allocated bandwidth.

Refer to "Calculating the values for WFQ source and destination-based traffic scheduling" for information on assigning *queue0-weight* to *queue3-weight* values.

Configuring maximum rate-based traffic scheduling

To configure maximum rate-based scheduling use a command such as the following.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# qos max-rate 100 100 100 100
```

Syntax: qos scheduler max-rate <Queue0-rate> <Queue1-rate> <Queue2-rate> <Queue3-rate>

The <Queue0-rate> variable defines the maximum bandwidth allocated to forwarding queue 0 in Kbps.

The <Queue1-rate> variable defines the maximum bandwidth allocated to forwarding queue 1 in Kbps.

The <Queue2-rate> variable defines the maximum bandwidth allocated to forwarding queue 2 in Kbps.

The <Queue3-rate> variable defines the maximum bandwidth allocated to forwarding queue 3 in Kbps.

Configuring minimum rate-based traffic scheduling

To configure minimum rate-based scheduling use a command such as the following.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# qos min-rate 100 100 100 100
```

Syntax: qos scheduler min-rate <Queue0-rate> <Queue1-rate> <Queue2-rate> <Queue3-rate>

The <Queue0-rate> variable defines the minimum bandwidth allocated to forwarding queue 0 in Kbps.

The <Queue1-rate> variable defines the minimum bandwidth allocated to forwarding queue 1 in Kbps.

The <Queue2-rate> variable defines the minimum bandwidth allocated to forwarding queue 2 in Kbps.

The <Queue3-rate> variable defines the minimum bandwidth allocated to forwarding queue 3 in Kbps.

Displaying the scheduler configuration

To view a Scheduler configuration, use the following command.

```
BigIron RX#show qos scheduler
Port   | Scheduler             Type   Prio0  Prio1     Prio2      Prio3
       | (Rates where specified are in Kbps)
-------+-------------------------------------+---------+---------+---------
13/1   | strict
13/2   | enhanced-strict       Rate   100000  200000    300000    Remaining
13/3   | min-rate              Rate   102400  204800    307200    409600
13/4   | strict
13/5   | strict
13/6   | max-rate              Rate   400000  400000    800000    10000000
13/7   | destination-weighted  Weight 15      25        25        35
13/8   | strict
13/9   | source-weighted       Weight 5       15        35        45
13/10  | strict
13/11  | strict
13/12  | strict
13/13  | strict
13/14  | strict
13/15  | strict
13/16  | strict
13/17  | strict
13/18  | strict
13/19  | strict
13/20  | strict
13/21  | strict
13/22  | strict
13/23  | strict
13/24  | strict
```

Syntax: show qos scheduler

Configuring multicast traffic engineering

Using the multicast traffic engineering feature, you can limit the amount of multicast traffic that passes through a packet processor. This command is configured on an individual port but applies to all ports connected to the same packet processor.

NOTE
If a user configures any one port between 1-12 (or 13 - 24) with a 'qos multicast best-effort rate of 1Mbps', no more then 1Mbps of multicast traffic will be forwarded at one time on ports 1-12 or (13-24).

Starting release 02.5.00, data-plane multicast traffic is rate-limited to 1.8 Gbps per packet processor.

NOTE
Using the **qos multicast best-effort rate** command affects data-plane (non-control protocol) multicast, broadcast and unknown unicast flooded traffic, that prior to inclusion of the command there was a potential for this traffic to starve other traffic from accessing an egress queue. The limiting on a per traffic manager basis to 1.8 Gbps was best for the majority of environments. Some high intensity multicast environments may need to increase this value to better match their network requirements.

To limit the multicast traffic through the packet processor that includes port 1/1 to 10 Mbps, use the following command.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# qos multicast best-effort rate 10000
```

Syntax: qos multicast best-effort rate <rate>

The <rate> variable defines the bandwidth of multicast traffic that is allowed to pass through the packet processor that include the port this command is configured on. On a 24-port x 1 Gbps Interface module, a qos multicast command applied to any of the ports numbered 1 to 12 will apply to all of these ports. Any command applied to ports numbered 13 to 24 will apply to these ports.

This variable is configured in Kbps.

The minimum configurable rate is 10 Mbps.

Displaying the multicast traffic engineering configuration

To view multicast traffic engineering configurations, use the following command.

```
BigIron RX#show qos multicast
Port    | Best Effort
        | Bandwidth (Kbps)
        |
--------+-------------------
13/1    | 140000
13/2    | 140000
13/3    | 140000
13/4    | 140000
13/5    | 140000
13/6    | 140000
13/7    | 140000
13/8    | 140000
13/9    | 140000
13/10   | 140000
13/11   | 140000
13/12   | 140000
13/13   | 12000000
13/14   | 12000000
13/15   | 12000000
13/16   | 12000000
13/17   | 12000000
13/18   | 12000000
13/19   | 12000000
13/20   | 12000000
```

```
13/21    | 12000000
13/22    | 12000000
13/23    | 12000000
13/24     | 12000000
```

Syntax: show qos multicast [<portnum>]

The <portnum> variable allow you to optionally limit the display to an individual port.

QoS for the oversubscribed 16 x 10GE modules

The 16-port 10 Gigabit Ethernet oversubscribed module plugs into any port slot of the device switch and is compatible with all previous generations of card on that switch. It provides interfaces to 16 X 10GE ports. This module will provide 4:1 oversubscription on the 16 x10GE network ports.

Aggregation NP QOS modes

The 16x10 module supports two ingress scheduling modes: Server and Storage.

The Aggregation NP has 4 group schedulers. Each scheduler support 4 ports. Each port is given a high and low priority queue.

In Server mode the group scheduler uses strict priority between the high and low priority queues. Scheduling within a priority is done with WRR using equal weights.

In Storage mode the group scheduler uses WRR to schedule all high and low priority queues. The weights are configured using a CLI command.

The 16-port 10 Gigabit Ethernet module works in Server mode by default.

Configuration considerations

- Mirror (analyzer) ports cannot be assigned to the 16x10GE card. You can monitor traffic on 16x10 ports.
- Brocade currently only support "strict" and "destination-weighted" scheduling schemes.
- Virtual interface subsets are not supported for engress ACLs.
- The egress filtering of the 16x10 module only compares to 3 bits of TOS field (delay, throughput, reliability).

Port group assignments

The 16 x10 GE module consists of 4 port groups of 4 ports each: Each of the network ports are assigned high low and low high queues.

- Port group 1: ports 1,5,9,13
- Port group 2: ports 2,6,10,14
- Port group 3: ports 3,7,11,15
- Port group 4: ports 4,8,12,16

Setting the server and storage modes

The default mode is **qos rcv-scheduler fq-sp** which is strict priority mode. This is used for the Server mode. This command sets the queues (TC) associated to the uplink ports. Each network port is assigned one low and high priority queue. In the either mode, queues 1-4 are low priority and 5-8 are high priority. For example, network port 1 uses queues (TC) 1 and 5. In the strict priority mode the scheduling between high and low priority is strict.

For both Server or Storage mode, the network control traffic will use Drop Precedence 0. (DP0) The incoming network control traffic will be assigned DP0 and all other traffic will be assigned DP1. This will allow the module to prefer network control during congestion conditions. Network control traffic will always be assigned to the high priority queue that is associated to the incoming network port.

Switching between server and storage modes

For Server mode, the default is **qos rcv-scheduler fq-sp** which is strict priority mode. This command sets the queues (TC) associated to the uplink ports. In either mode, queues 1-4 are low priority and 5-8 are high priority. Each network port is assigned one low and high priority queue. For example, network port 1 uses queues (TC) 1 and 5. In the strict priority mode the scheduling between high and low priority is strict. To enable the fair queuing strict priority mode, enter a command such as the following.

```
BigIron RX(config-if-e10000-4/2)#qos rcv-scheduler fq-sp
```

Qos profiles

The 16 x 10GE module uses QOS profiles to define the QOS treatment applied to packets. Each 16 x 10GE module has 16 different QOS profiles that are used on the traffic coming from the network ports. The profiles include traffic class and drop precedence. Each port will have the following QOS profiles.

1. TCx (low priority), DP1
2. TCx (low priority), DP0
3. TCx (high priority), DP1
4. TCx (high priority), DP0

Table 94 represents the QOS profiles required for the ingress direction.

TABLE 94 QOS profile table

Index	TC	DP	Associated port (Network Port 1 = 0)	QOS profile
0	0	1	0 or 4	Low priority TC DP1 (default)
1	1	1	1 or 5	Low priority TC DP1 (default)
2	2	1	2 or 6	Low priority TC DP1 (default)
3	3	1	3 or 7	Low priority TC DP1 (default)
4	0	0	0 or 4	Low priority TC DP0
5	1	0	1 or 5	Low priority TC DP0
6	2	0	2 or 6	Low priority TC DP0

TABLE 94 QOS profile table (Continued)

7	3	0	3 or 7	Low priority TC DP0
8	4	1	0 or 4	High priority TC DP1
9	5	1	1 or 5	High priority TC DP1
10	6	1	2 or 6	High priority TC DP1
11	7	1	3 or 7	High priority TC DP1
12	4	0	0 or 4	High priority TC DP0 (Network control)
13	5	0	1 or 5	High priority TC DP0 (Network control)
14	6	0	2 or 6	High priority TC DP0 (Network control)
15	7	0	3 or 7	High priority TC DP0 (Network control)

Setting the group port weights

The command **qos rcv-scheduler wfq 1 2 3 4 5 6 7 8 1 2 1 2 1 2** is used for Storage mode. This mode uses weighting to determine the queue scheduling. The network ports are still assigned high low and low high queues but are scheduled based on the weights assigned.

Setting the averaging-fair-weight (wfq) parameter

The **wfq** parameter is configured as the **averaging-fair-weight** parameter. In this implementation, you can set one of 13 (1 - 13) possible values. These values represent a wfg value as described in Table 95

Calculating the values for WFQ storage mode traffic scheduling

Weighted Fair Queueing (WFQ) scheduling is configured to be a percentage of available bandwidth using the following formula.

$$\text{Weight of } w(x) = \frac{w(x)}{w0 + w1 + w2 + w3 + w4 + w5 + w6 + w7}$$

Where.

w (x) = The value of the queue that you want to determine the weight for. It can be the value of any weight (0 - 7).

w0 - w7 = the assigned values of the eight weights.

Weight of w (x) = the calculated weight as a percentage of the port's total bandwidth.

For example if you assign the following values to weight 0 to 7.

```
BigIron RX (config-if-e10000-4/1)#qos rcv-scheduler wfq 1 5 1 5 1 5 1 5
```

Weight 0 = 1, Weight 1 = 5, Weight 2 = 1, Weight 3 = 5, Weight 4 = 1, Weight 5 = 5, Weight 6 = 1, and Weight 7 = 5

To determine the weight of w3.

$$\text{Weight of w3} = \frac{5}{1 + 5 + 1 + 5 + 1 + 5 + 1 + 5}$$

The weight of w3 is 20.8%. Consequently, w3 (Port 2 High Priority) will get 20.83% of the group port's total bandwidth if equal amounts of traffic are received from all eight weights.

The values of the remaining weights are calculated to be the following: w0 = 4.17%, w1 = 20.83%, w2 = 4.17%, w4 = 4.17%, w5 = 20.83%, w6 = 4.17%, and w7 = 20.83%

Egress port shaping

The 16x10GE module is designed to provide port fairness, but the cost is a smaller number of usable queues per input port (on egress). Traffic received on a network port will be assigned to one of 2 egress queues with a specific drop precedence value.

Table 95 identifies the profile used for network control traffic which is identified using an independent flag.

TABLE 95 QOS profile index

Qos profile	QOS profile index (depending on network port)	Comments
Low priority DP1	0,1,2,3	
Low priority DP1	0,1,2,3	
Low priority DP1	0,1,2,3	
Low priority DP0	4,5,6,7	
Low priority DP0	4,5,6,7	
High priority DP1	8,9,10,11	
High priority DP1	8,9,10,11	
High priority DP1	8,9,10,11	
High priority DP0	12,13,14,15	Only used for control traffic

Mirroring ports

The 16x 10GE module supports mirroring, but with the following limitations:

- A 16X10GE port cannot be configured as mirror port.
- Only one port can be monitored at any time from ports 1 - 8, and one port can also be monitored at any time from ports 9 - 16.
- The mirror port for ingress or egress should be the same port, they cannot be monitored on multiple separate mirror ports.
- Mirror(analyzer) ports cannot be assigned to the 16x10GE module. You can monitor traffic on 16x10 ports.

Supported ACLs

The 16x10GE module supports standard, extended, named and numbered egress ACLs. Refer to Chapter 21, "Access Control List" for additional information.

Configuring QoS for the 16 x 10G module

New CLI commands have been added to allow alternating between server and storage modes on the 10 x 16GE module. The new commands are part of the **qos** group, and configured at the interface level.

Configuration steps

1. To set the group port 1 weight, low prioriy traffic, enter the following command.

```
BigIron RX(config-if-e10000-4/1)#qos rcv-scheduler wfq 1
```

2. To set the group port 1 weight, high prioriy traffic, enter the following command.

```
BigIron RX(config-if-e10000-4/1)#qos rcv-scheduler wfq 1 2
```

NOTE
The configurations for group port 1 will now be associated to s/1,s/5,s/9,s/13

3. To set the group port 2 weight, low prioriy traffic, enter the following command.

```
BigIron RX(config-if-e10000-4/1)#qos rcv-scheduler wfq 1 2 1
```

4. To set the group port 2 weight, high prioriy traffic, enter the following command.

```
BigIron RX(config-if-e10000-4/1)#qos rcv-scheduler wfq 1 2 1 2
```

NOTE
The configurations for group port 2 will now be associated to s/2,s/6,s/10,s/14

5. To set the group port 3 weight, low prioriy traffic, enter the following command.

```
BigIron RX(config-if-e10000-4/1)#qos rcv-scheduler wfq 1 2 1 2 1
```

6. To set the group port 3 weight, high prioriy traffic, enter the following command.

```
BigIron RX(config-if-e10000-4/1)#qos rcv-scheduler wfq 1 2 1 2 1 2
```

NOTE
The configurations for group port 3 will now be associated to s/3,s/7,s/11,s/15

7. To set the group port 4 weight, low prioriy traffic, enter the following command.

```
BigIron RX(config-if-e10000-4/1)#qos rcv-scheduler wfq 1 2 1 2 1 2 1
```

8. To set the group port 4 weight, high prioriy traffic, enter the following command.

```
BigIron RX(config-if-e10000-4/1)#qos rcv-scheduler wfq 1 2 1 2 1 2 1 2
```

NOTE
The configurations for group port 4 will now be associated to s/4,s/8,s/12,s/16

Syntax: [no]qos rcv-scheduler fq-sp l wfg [num]

Use **rcv-scheduler** to change the receive scheduling parameters on the 16x10G card.

Use **scheduler** to assign a scheduling mechanism to one or more ports.

Use the **no** parameter to return to the default mode. (Server)

Use the **fq-sp** parameter to set the 16x10G module to fair queuing strict priority mode.

Use the **wfq** parameter to set the 16x10G module to weighted fair queuing mode.

Use the **num** parameter to set the port weight. Refer to Table 95 on page 491 for additional information on possible values.

The **no qos rcv-scheduler** command is used to return to the default mode (Server).

Chapter

19 Configuring Traffic Reduction

In this chapter

Traffic policing on the BigIron RX Series

The BigIron RX Series Router provides line-rate traffic policing in hardware on inbound ports and outbound ports.

You can configure a BigIron RX Series 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. These policies can be applied to inbound and outbound traffic. (Refer to "Configuring a port-based rate limiting policy" on page 498.)
- **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. These policies can be applied to inbound and outbound traffic.
- **Port-and-VLAN-based** – Limits the rate of packets tagged with a specific VLAN on an individual physical port. Only one rate can be specified for each VLAN.
- **VLAN-group-based** – Limits the traffic for a group of VLANs. Members of a VLAN group share the specified bandwidth defined in the rate limiting policy that has been applied to that group. You can configure multiple VLAN group rate limits. Each grouping of Port + VLAN Groups will take up multiple entries from the CAM (one entry for each VLAN in the group).
- **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). 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. (Refer to "Configuring a port-and-ACL-based traffic policing policy" on page 501.)
- **Port-and-IPV6 ACL-based** – Limits the rate of traffic on an individual physical port that matches the permit conditions of IPV6 ACL. These policies can be applied to inbound traffic only. (Refer to "Configuring a port-and-IPv6 ACL-based traffic reduction" on page 502.)

Traffic reduction parameters and algorithm

A rate limiting policy specifies two parameters: requested rate and maximum burst.

Requested rate

The *requested 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 rate limiting policy will not exceed the requested rate.

The requested rate represents a percentage of an interface's line rate (bandwidth), expressed in bits per second (bps).

Requested Rate must be entered in multiples of 515,624 bps. If you enter a number that is not a multiple of 515,624, 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 600,000 bps, the value will be adjusted to 515,624 bps.

Maximum burst

Maximum burst provides a higher than requested rate to traffic that meet the rate limiting criteria. When the traffic on the port is less than the specified requested rate, the rate limiting policy can accumulate credits up to a maximum, as specified in the maximum burst value. The accumulated credit allows traffic to pass through the port for a short period of time, at a rate higher than the average rate. The time period is determined by the amount of credit accumulated and the rate of traffic passing through the port.

The maximum burst rate cannot be smaller than 65536 bits

Actual rate

The device determines *actual* rate limiting rates through the use of proprietary formulas built into the packet processor hardware. The resulting rate that is the closest to the requested rate. This leads to variable rate limiting granularities for rate limiting rates. The lower the configured rate limiting rate, the finer the granularity; the higher the configured rate limiting rate, the larger the rate increments. For example, at lower rates, say from 20,345 to 40, 330 bps, the configurable rate limiting rates can increment by 1 bps, but at higher rate limiting rates, say between 4 Gbps to line-rate, the configurable rates increment in the hundreds of Mbps.

The CLI shows actual rate as *requested rate.*

Credits and credit total

Each rate limiting policy is assigned a class. A *class* uses the average rate and maximum allowed burst in the rate limiting policy to calculate credits and credit totals.

Credit size is measured in bytes. A credit is a forwarding allowance for a rate-limited port, and is the smallest number of bytes that can be allowed during a rate limiting interval. The 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.

The credit size is calculated using the following algorithm.

Credit = (Average rate in bits per second)/(8*64453)

One second is divided into 64,453 intervals. 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 calculated using the following algorithm.

Maximum credit total = (Maximum burst in bits)/8

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 rate limiting 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 rate limiting criteria, then the running total can increase until it reaches the maximum credit total.

Configuration considerations

- Except for port-based rate limiting policies, all rate limiting policy types can be applied only to inbound ports on the device.
- Only one type of inbound rate limiting can be applied on a physical port. For example, you cannot apply inbound port-and-ACL-based and inbound port-based rate limiting policies on the same port.
- Outbound port-based rate limiting policy can be combined with any type of inbound rate limiting policy.
- Any VLAN-based rate limiting can limit only tagged packets that match the VLAN ID specified in the policy. Untagged packets are not subject to rate limiting.
- The minimum configurable rate limiting rate in the device is 20,345 bps. The maximum configurable rate limiting rate is near line-rate.
- Control packets are not subject to rate limiting.
- Trunks with rate-limiting on their physical members is configurable where all the ports inherit the rate-limiting policy.
- The device currently only support "strict" and "destination-weighted" scheduling schemes. (qos scheduler ..) on the 16x10G card.
- Certain features such as FDP, CDP, UDLD and LACP that make the port run in dual mode can cause traffic to be rate limited to less than the expected requested rate. When the port is in dual-mode, all incoming or outgoing packets are treated as tagged. An extra 4 bytes is added to the length of the packet to account for the tag, thus causing the requested rate to be less than the expected requested rate. Ports in dual mode are assumed to be tagged ports for rate limiting purpose.
- The CAM can hold up to 1024 ACL, PBR, and Rate Limiting entries and this maximum is divided as follows:
  - ACL – 416 entries
  - Rate Limiting – 416, entries shared with PBR

  Also note:
  - Port-based and VLAN-based rate limiting policies consume one CAM entry per policy.

- ACL-based rate limiting policies consume entries based on the number of statements in an ACL.
- See the limits in Table 96.

TABLE 96 Maximum # of rate limiting policies and VLANs w/ byte accounting permitted per-PPCR

Module type	PPCR number	Port #	Max # of rate limiting policies based on ACLs and VLANs + number of VLANs w/ byte accounting enabled
4 x 10G	PPCR 1	1	126
	PPCR 2	2	126
	PPCR 3	3	126
	PPCR 4	4	126
24 x 1G	PPCR 1	1 - 12	115
	PPCR 2	13 - 24	115

Configuring rate limiting policies

Configuring rate-limiting policies involves using the rate-limit command and specifying the requested rate and maximum burst at the interface level of the CLI.

Software release 02.4.00 adds rate limiting to outbound ports.

Configuring a port-based rate limiting policy

Only one port-based rate limiting policy can be applied to an inbound port.

To configure a port-based rate limiting policy, enter commands such as the following.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# rate-limit out 500000000 750000000
Average rate is adjusted to 499639656 bits per second
```

The commands configure a rate limiting policy for outbound traffic on port 1/1. The policy requests to limit the rate on all inbound traffic to 500 Mbps with a maximum burst size of 750 Mbps. The device adjusts the requested rate to 499639656 bits per second.

Syntax: [no] rate-limit input | output <requested-rate> <maximum-burst>

input applies rate limiting to inbound traffic on the port. **Input** can be abbreviated as **in**.

Output applies rate limiting to outbound traffic on the port. **Output** can be abbreviated as **out**.

The <requested-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The minimum configurable requested rate is 20,345 bps. The maximum configurable rate limiting rate is near line-rate.

Refer to "Requested rate" on page 496 for more details.

The <maximum-burst> parameter specifies the extra bits above the requested rate that traffic can have. Refer to "Maximum burst" on page 496 for more details.

Configuring a port-and-priority-based rate limiting policy

802.1p packet priority is used by default. The priority number specifies the IEEE 802.1 equivalent to one of the four Brocade QoS queues. You can configure port-and-priority-based rate limiting for each of the priority numbers 1 - 7 on a port.

To configure a port-and-priority-based rate limiting policy, enter commands such as the following at the interface level.

```
BigIron RX(config)# interface ethernet 1/2
BigIron RX(config-if-e1000-1/2)# rate-limit in priority 0 500000000 750000000
Average rate is adjusted to 499639656 bits per second
BigIron RX(config-if-e1000-1/2)# rate-limit in priority 1 priority 2 priority 3
650000000 650000000
```

The commands configure port-and-priority-based rate limiting policies on inbound port 1/2. The policies requests a rate limit of 500 Mbps on hardware forwarding queues 0 and 1 on the port, with a maximum burst size of 750 Mbits. The device adjusts the requested rate to 499639656 bits per second.

Syntax: [no] rate-limit input priority <num> <requested-rate> <maximum-burst>

The **priority** <num> parameter specifies the 802.1p priority levels 0 – 7, equivalent to one of the four QoS queues. For information on the priority level and the corresponding queue, refer to "Assigning QoS priorities to traffic" on page 469.

For information on the other parameters, refer to "Configuring a port-based rate limiting policy" on page 498.

Configuring a port-and-VLAN-based rate limiting policy

To configure a port-and-VLAN-based rate limiting policy, enter commands such as the following.

```
BigIron RX(config)# interface ethernet 1/3
BigIron RX(config-if-e1000-1/3)# rate-limit in vlan 10 500000000 750000000
Average rate is adjusted to 499321856 bits per second
BigIron RX(config-if-e1000-1/3)# rate-limit in vlan 20 100000000 600000000
Average rate is adjusted to 97523712 bits per second
```

The commands configure two rate limiting policies that limit the requested rate of inbound traffic on port 1/3. The first policy request to limit packets with VLAN tag 10 to an rate of 500 Mbps with a maximum burst size of 750 Mbits on the port. The second policy requests to limit packets with VLAN tag 20 to an rate of 100 Mbps with a maximum burst size of 600 Mbits on the port. Tagged packets belonging to VLANs other than 10 and 20 and untagged packets are not subject to rate limiting on port 1/3.

Syntax: [no] rate-limit input vlan <vlan-number> <requested-rate> <maximum-burst>

The vlan <vlan-number> parameter species the VLAN ID to which the policy applies. Refer to "Configuration considerations" on page 497 to determine the number of rate limiting policies that can be configured on a device.

For information on the other parameters, refer to "Configuring a port-based rate limiting policy" on page 498.

Configuring a VLAN-group-based rate limiting policy

A rate limiting policy can be applied to a VLAN group. VLANs that are members of a VLAN group share the specified bandwidth defined in the rate limiting policy applied to that group.

To configure a rate limiting policy for a VLAN group, do the following.

1. Define the VLANs that you want to place in a rate limiting VLAN group.
2. Define a rate limiting VLAN group (it is specific to the rate limiting feature) and assign VLANs to it. To define a rate limiting VLAN group, use the **rl-vlan-group** command at the CONFIG level. To assign VLANs to the group, use the **vlan** command at the VLAN group rate limiting configuration level.

   For example, enter the following.

   ```
   BigIron RX(config)# rl-vlan-group 10
   BigIron RX(config-rl-vlan-group-10)# vlan 3 5 to 7
   BigIron RX(config-rl-vlan-group-10)# exit
   ```

   The commands assign VLANs 3, 5, 6, and 7 to rate limiting VLAN group 10.

   Syntax: [no]rl-vlan-group <vlan-group-number>

   Syntax: [no]vlan <vlan-number> [to <vlan-number]

   The **rl-vlan-group** command defines a rate limiting VLAN group and takes you to the VLAN group rate limiting configuration level.

   <vlan-group-number> specifies the VLAN group that you want to create.

   The **vlan** command assigns VLANs to the rate limiting VLAN group. Possible values are individual VLAN IDs or a range of VLAN IDs.
3. Create a rate limiting policy for the VLAN group and apply it to the interface. Enter the command such as the following at the interface level.

   ```
   BigIron RX(config-if-e1000-1/4)# rate-limit in group 10 500000000 750000000
   ```

   The command configures a rate limiting policy on port 1/4 that limits the rate of inbound traffic (packets tagged with VLANs 3, 5, 6, or 7 from VLAN group 10) from VLAN group 10 to 500 Mbps with a maximum burst size of 750 Mbits.

   Syntax: rate-limit in group <group-number> <requested-rate> <maximum-burst>

   The **group** <group-number> parameter specifies the rate limiting VLAN group.

   For information on the other parameters, refer to "Configuring a port-based rate limiting policy" on page 498.
4. To apply a rate limiting policy to a VLAN group whose traffic is prioritized by hardware forwarding queues, enter the command such as the following in lieu of step number 3.

   ```
   BigIron RX(config-if-e1000-1/4)# rate-limit in group 10 priority 5 priority 6
   500000000 750000000
   ```

   The command applies the rate limiting policy for rate limiting VLAN group 10. This policy limits all traffic tagged with VLANs 3, 5, 6, or 7 on hardware forwarding queues 2 and 3 to a rate of 500 Mbps with a maximum burst size of 750 Mbits.

   Syntax: rate-limit in group <group-number> priority <num> <requested-rate> <maximum-burst>

The **priority** <num> parameter specifies the 802.1p priority levels 0 - 7, equivalent to one of the four QoS queues. For information on the priority levels and the corresponding queue, refer to "Assigning QoS priorities to traffic" on page 469.

For information on the requested rate and maximum burst, refer to "Configuring a port-based rate limiting policy" on page 498.

Configuration considerations for VLAN-group-based rate limiting policies

When configuring VLAN group based rate limiting policies, consider the following rules:

- A rate limit VLAN group must have at least one VLAN member before it can be used in a rate limit policy. The list cannot be empty if it is being used in a rate limiting policy.
- A rate limit VLAN group cannot be deleted if it is being used in a rate limiting policy.
- If a rate limit policy for a VLAN group is applied to a port, the group cannot be used in any other rate limiting policies applied to other ports that are controlled by the same packet processor.
- A VLAN can be member of multiple rate limit VLAN groups, but two groups with common members cannot be applied on ports controlled by the same packet processor.
- VLAN-based rate limiting and VLAN groups based rate limiting policies can be applied on the same ports or ports controlled by the same packet processor as long as there are no common VLANs in the policies.

Configuring a port-and-ACL-based traffic policing policy

You can use standard or extended ACLs for port-and-ACL-based rate limiting policies:

- 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 apply an ACL ID to a port-and-ACL-based rate limiting policy before you define the ACL. The rate limiting 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.

Refer to the Chapter 21, "Access Control List" for details on how to configure ACLs.

To configure a port-and-ACL-based rate limiting policy, enter commands such as the following.

```
BigIron RX(config)#access-list 50 permit host 1.1.1.2
BigIron RX(config)#access-list 50 deny host 1.1.1.3
BigIron RX(config)#access-list 60 permit host 2.2.2.3
BigIron RX(config)#int e 1/5
BigIron RX(config-if-e1000-1/5)# rate-limit in access-group 50 500000000
750000000
Average rate is adjusted to 499321856 bits per second
BigIron RX(config-if-e1000-1/5)# rate-limit in access-group 60 100000000
200000000
Average rate is adjusted to 97523712 bits per second
```

These commands first configure access-list groups that contain the ACLs that will be used in the rate limiting policy. Use the **permit** condition for traffic that will be rate limited. Traffic that match the condition are not subject to rate limiting and allowed to pass through. Refer to "Configuring a port-and-IPv6 ACL-based traffic reduction" on page 502 for information on how to drop traffic that matches deny conditions.

Next, the commands configure two rate limiting policies on port 1/5. The policies limit the 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 from host 1.1.1.2 to an requested rate of 500 Mbps with a maximum burst size of 750 Mbps. 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 requested 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 rate limiting.

Syntax: [no] rate-limit in access-group <number> | named-access-group <ACL-name> <requested-rate> <maximum-burst>

The **access-group** <number> parameter or the **named-access-group** <acl-name> specifies the ACL used in the policy.

For information on the other parameters, refer to "Configuring a port-based rate limiting policy" on page 498.

For information on the number of ACL-based rate limiting policies that can be configured, refer to the "Configuration considerations" on page 497.

Configuring a port-and-IPv6 ACL-based traffic reduction

The port-and-IPV6 ACL-based rate limiting limits the rate of traffic on individual physical ports that match the permit conditions of an IPV6 ACL. Traffic that matches the deny condition is not subject to rate limiting.

For example, the following commands in the Global Config mode configure the IPv6 access-list "sample" to permit any traffic from the 10:10::0:0/64 network and deny all other traffic.

```
BigIron RX(config)# ipv6 access-list sample
BigIron RX(config-ipv6-access-list sample)# permit ipv6 10:10::0:0/64 any
BigIron RX(config-ipv6-access-list sample)# deny ipv6 any any
```

The following configuration creates a rate limiting policy on port 1/1. The policy limits the rate of all inbound IP traffic that matches the permit rules a rate of 100 Mbps with a maximum burst size of 200 Mbits. Traffic denied by sample is forwarded on the port.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e10000-1/1)# rate-limit in ipv6-named-access-group sample
100000000 200000000
Average rate is adjusted to 99515432 bits per second
```

Syntax: [no] rate-limit in ipv6-named-access-group <name> <requested-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The **ipv6-named-access-group** <name> parameter identifies the IPv6 ACL used to permit or deny traffic on a port. Permitted traffic is subject to rate limiting. Denied traffic is forwarded on the port.

For information on the other parameters, refer to "Configuring a port-based rate limiting policy" on page 498.

NP based multicast, broadcast, and unknown-unicast rate limiting

NOTE
Beginning with release 02.7.00, the **multicast limit**, **broadcast limit**, and the **unknown-unicast limit** commands have been supperseded with the **multicast rate-limit**, **broadcast rate-limit**, and the **unknown-unicast rate-limit** commands. You must reconfigure the rate limiting when upgrading to the 02.7.00 Multi-Service Ironware.

Beginning with release 02.7.00 of the Multi-Service IronWare, NP based Multicast, Broadcast, and Unknown-Unicast rate limiting is supported. This feature allows hardware NP based Multicast, Broadcast, and Unknown-Unicast rate limiting for both CPU based flooding and hardware based flooding.

To enable multicast rate-limiting on a specific port, enter a command such as the following.

```
BigIron RX (config)# multicast rate-limit 1000000 1 np 2/2
```

Syntax: [no] multicast rate-limit <avg-rate> <max-burst> np [slot/port | all]

```
To enable Broadcast rate-limiting on a specific port, enter a command such as the
following.
BigIron RX (config)# broadcast rate-limit 1000000 1 np 3/2
```

Syntax: [no] broadcast rate-limit <avg-rate> <max-burst> np [slot/port | all]

```
To enable unknown unicast rate-limiting on a specific port, enter a command such
as the following:
BigIron RX (config)# unknown-unicast rate-limit 1000000 1 np 4/2
```

Syntax: [no] unknown-unicast rate-limit <avg-rate> <max-burst> np [slot/port | all]

Use the **no** parameter to disable np rate limiting.

The <**avg-rate**> is entered in Kbits/sec. Possible values are from 1-429467295. The default value is 4294967295.

The <**max-burst**> parameter specifies the total number bits that can pass during a burst. Possible values are from 1-429467295. The default value is 4294967295.

The **np** parameter specifies the rate limit per network processor.

The **slot/port** specifies the interface module and port to be rate limited.

The **all** parameter specifies that you want all the ports to be rate limited.

NOTE
When you specify default values for both average rate and burst size, the values will not be displayed in the **show running configuration** output.

Displaying traffic reduction

The **show rate-limit** command displays the rate limiting policies configured on the ports. For example.

```
BigIron RX(config)# show rate-limit
interface e 1/1
 rate-limit input 499321856 750000000
 interface e 1/3
 rate-limit input vlan-id 10 499321856 750000000
 rate-limit input vlan-id 20 97523712 200000000
```

To display bytes forwarded and dropped, enter the following command.

```
BigIron RX(config)# show rate-limit counters
interface e 1/1
 rate-limit input 499321856 750000000
 Bytes fwd: 440 Bytes drop: 20 Total: 460
interface e 1/3
 rate-limit input vlan-id 10 499321856 750000000
 Bytes fwd: 0 Bytes drop: 0 Total: 0
rate-limit input vlan-id 20 97523712 200000000
 Bytes fwd: 0 Bytes drop: 0 Total: 0
```

The byte count includes the preamble and the minimum inter-frame gap in Ethernet.

To display rate limiting policies for an interface that includes counters, enter the following command.

```
BigIron RX(config)# show rate-limit counters interface 1/1
interface e 1/1
 rate-limit input 499321856 750000000
 Bytes fwd: 440 Bytes drop: 20 Total: 460
```

To display the rate limiting policies on interface 1/3, enter the following command.

```
BigIron RX(config)# show rate-limit interface 1/3
interface e 1/3
rate-limit input vlan-id 10 499321856 750000000
 rate-limit input vlan-id 20 97523712 200000000
```

To display rate-limit VLAN groups, enter the following.

```
BigIron RX(config)# show rate-limit group
rl-vlan-group 10
 vlan 3 5 to 7
```

Syntax: show rate-limit [counters [interface <slot/port>]] [group [<vlan-number>]] [interface <slot/port>]

The **counters** parameter displays bytes forwarded and dropped by the interfaces that have a rate-limiting policy. <slot/port> specifies a particular interface.

The **group** <vlan-number> parameter indicates the rate limiting VLAN group for which the rate-limiting policy is created.

interface <slot/port> displays the rate limiting policy for a particular interface.

Chapter

Layer 2 ACLs 20

In this chapter

This chapter presents information to configure and view Layer 2 ACLs.

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

The Layer 2 ACL feature is unique to Brocade devices and differs from software-based MAC address filters. MAC address filters use the CPU to filter traffic; therefore, performance is limited by the CPU's processing power. Layer 2 ACLs filter traffic at line-rate speed.

Filtering based on ethertype

Layer 2 ACLs can filter traffic based on protocol type. For each Layer 2 ACL etype entry bound to a port, a CAM entry is written to the corresponding CAM. You can conserve CAM space by configuring only the Layer 2 ACLs needed. For instance, to filter only IPV4-Len-5 traffic, specify that particular etype. This results in one CAM entry. Configuration examples are provided in the section "Configuring Layer 2 ACLs" on page 506

You can configure Layer 2 ACLs to use the **etype** argument to filter on the following etypes:

- IPv4-Len-5 (Etype=0x0800, IPv4, HeaderLen 20 bytes)
- ARP (Etype=0x0806, IP ARP)
- IPv6 (Etype=0x86dd, IP version 6)

Configuration rules and notes

- You cannot bind Layer 2 ACLs and IP ACLs to the same port. However, you can configure one port on the device to use Layer 2 ACLs and another port on the same device to use IP ACLs.
- You cannot bind a Layer 2 ACL to a virtual interface.

- The Layer 2 ACL feature cannot perform SNAP and LLC encapsulation type comparisons.
- device processes ACLs in hardware.
- You can use Layer 2 ACLs to block management access to the device. For example, you can use a Layer 2 ACL clause to block a certain host from establishing a connection to the device through Telnet.
- 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.

Configuring Layer 2 ACLs

Configuring a Layer 2 ACL is similar to configuring standard and extended ACLs. Layer 2 ACL table IDs range from 400 to 499, for a maximum of 100 configurable 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.

The device evaluates traffic coming into the port against each ACL clause. When a match occurs, the device takes the corresponding action. Once a match entry is found, the device either forwards or drops the traffic, depending upon the action specified for the clause. Once a match 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 and assumes you know what you are doing.

Creating a Layer 2 ACL table

You create a Layer 2 ACL table by defining a Layer 2 ACL clause.

To create a 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.

```
BigIron RX(config)# access-list 400 deny any any any etype arp
BigIron RX(config)# access-list 400 deny any any any etype ipv6
BigIron RX(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, refer to "Example Layer 2 ACL clauses" on page 507.

Syntax: [no] access-list <*num*> permit | deny <*src-mac*> <*mask*> | any <*dest-mac*> <*mask*> | any [<*vlan-id*> | any [etype <*etype-str*>] [log-enable]]

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 F's 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 do not 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-id of 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*> can be one of the following keywords:

- IPv4-Len-5 (Etype=0x0800, IPv4, HeaderLen 20 bytes)
- ARP (Etype=0x0806, IP ARP)
- IPv6 (Etype=0x86dd, IP version 6)

The optional <*log-enable*> parameter enables the logging mechanism. The device accepts this command only when a **deny** clause is configured. When you enable logging for a Layer 2 ACL, all traffic matching the clause is sent to the CPU for processing and traffic is denied by the CPU. The CPU creates a log entry for the first packet that is denied and once every 10 seconds thereafter. The logging mechanism includes sending SNMP traps and log messages to the Syslog servers and writing the log entry to the log buffer on the device.

In addition, if specified with a 'permit' action, the log-enable keyword is ignored and the user is warned that he cannot log permit traffic.

NOTE
Traffic denied by the implicit deny mechanism is not subject to logging. The implicit deny mechanism kicks in when the traffic does not match any of the clauses specified and there is no **permit any any** clause specified at the end.

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 Layer 2 ACL clauses

The following shows some examples of valid Layer 2 ACL clauses.

```
BigIron RX(config)# access-list 400 permit any any
BigIron RX(config)# access-list 400 permit any any log-enable
BigIron RX(config)# access-list 400 permit any any 100
BigIron RX(config)# access-list 400 permit any any 100 log-enable
BigIron RX(config)# access-list 400 permit any any any
BigIron RX(config)# access-list 400 permit any any any log-enable
BigIron RX(config)# access-list 400 permit any any 100 etype ipv4
BigIron RX(config)# access-list 400 permit any any 100 etype ipv4 log-enable
```

The following shows an example of a valid Layer 2 ACL clause.

```
BigIron RX(config)# access-list 400 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.

Binding a Layer 2 ACL table to an interface

To enable Layer 2 ACL filtering, bind the Layer 2 ACL table to an interface.

NOTE
Layer 2 ACLs cannot be bound to virtual routing interfaces.

Enter a command such as the following at the Interface level of the CLI.

```
BigIron RX(config)# int e 4/12
BigIron RX(config-int-e100-4/12)# mac access-group 400 in
```

Syntax: [no] mac access-group <*num*> in

The <*num*> parameter specifies the Layer 2 ACL table ID to bind to the interface.

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. You can specify a maximum of 256 clauses per table. The default value is 64 clauses per 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.

```
BigIron RX(config)# system-max l2-acl-table-entries 200
```

Syntax: system-max l2-acl-table-entries <*max*>

The <*max*> parameter specifies the maximum number of clauses per Layer 2 ACL. Enter a value from 64 to 256.

Viewing Layer 2 ACLs

Use the **show access-list** command to monitor configuration and statistics and to diagnose Layer 2 ACL tables. The following shows an example output.

```
BigIron RX(config)# show access-list 400
L2 MAC Access List 400:
  permit any any 100 etype ipv4
  deny any any any etype arp
```

Syntax: show access-list <*number*>

The <*num*> parameter specifies the Layer 2 ACL table ID.

Example of Layer 2 ACL deny 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.

```
BigIron RX(config)# access-list 401 deny 0012.3456.7890 ffff.ffff.ffff
0011.2233.4455 ffff.ffff.ffff
BigIron RX(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.

```
BigIron RX(config)# access-list 401 deny 0012.3456.7890 ffff.ffff.fffe
0011.2233.4455 ffff.ffff.ffff
BigIron RX(config)# access-list 401 permit any any
```

Chapter

21

Access Control List

In this chapter

This chapter describes the IP 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 "Types of IP ACLs" on page 513.

You can use IP 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. Also, if you use an ACL in a route map and you use a wildcard character as the source IP address, make sure you apply the route map to interfaces instead of globally, to prevent loops. See the chapters for a specific feature for information on using ACLs as input to those features.

How the device processes ACLs

The device processes traffic that ACLs filter in hardware. The device creates an entry for each ACL in the Content Addressable Memory (CAM) at startup or when the ACL is created. The device uses these CAM entries to permit or deny packets in the hardware, without sending the packets to the CPU for processing.

General configuration guidelines

- ACLs are supported on physical interfaces, trunk groups, and virtual routing interfaces.
- ACLs are supported only for inbound traffic. An error message is displayed if you apply an ACL to an outbound interface.
- You can create up to 416 CAM entries, but you can have up to 8,000 statements (rules) in all the ACL configurations on the device. Default is 4096 statements.
- A port supports only one ACL; however, the ACL can contain multiple statements. For example, both ACLs 101 and 102 cannot be supported on port 1, but ACL 101 can contain multiple entries.
- 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 551.
- You cannot enable any of the following features on the interface if an ACL is already applied to that interface:
  - Protection against ICMP or TCP Denial-of-Service (DoS) Attacks
  - ACL-based rate limiting
  - ACL Logging
  - Policy-based routing (PBR)

RX-BI-16XG (16 x 10GE) Module EGRESS ACLconfiguration guidelines

- The RX-BI-16XG 16 x 10GE module only supports standard, extended, named, and numbered ACLs for outbound access-group applicationsACLs.
- Egress filtering on subset ports of a VE is not supported, matching must apply to all VE ports .
- Matching the SPI field value is not supported for egress acl.
- Matching field of fragment or fragmentation-offset is not supported.
- A matching egress acl only compares to 3 bits of TOS field (delay, throughput, reliability)
- ACLs that specify spi, .tos min monrtary cost, fragment or fragmentation-offset will cause a configuration conflict and an error message "ACL configuration conflict specified filter not supported" is entered in syslog.
- 802.1p-priority is not supported as a matching egress acl condition.
- dscp-marking is not available as a condition matching egress acl action.
- deny-logging is not supported for egress ACLs.

Disabling or re-enabling Access Control Lists (ACLs)

The ACL feature is always enabled on device; it cannot be disabled.

Default ACL action

The default action when no ACLs are configured on a device 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:

- To control access more tightly, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.
- To secure access in environments with many users, you can 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, extended, or super. A standard ACL permits or denies packets based on a source IP address. An extended ACL permits or denies packets based on source and destination IP addresses and also based on IP protocol information. Super ACLs can match on any field in a packet header from Layer 2 to Layer 4. Super ACLs support all options currently supported in ACL and MAC ACL, including QoS marking.

Standard or extended ACLs can be numbered or named. Standard ACLs are numbered from 1 – 99, extended ACLs are numbered 100 – 199. Super ACLs may be assigned numbered IDs only, from 500 - 599. IDs for standard or extended ACLs can also be a character string (named). 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 ACL ID is a number from 1 – 99 (standard), 100 – 199 (extended) or 500 – 599 (super) or a character string (super ACLs are numbered only). 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 it easier to apply large groups of access filters (ACL entries) to interfaces.

  NOTE
  This process differs from the process of assigning IP access policies. When you use IP access policies, you apply the individual policies directly to the interfaces.

- **ACL entry** – An ACL entry contains 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 device 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 traffic on specific ports. You can apply only one ACL to a port's inbound 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 (for example, permit or deny the packet) and stops further comparison for that packet.

Enabling support for additional ACL statements

You can enable support for additional ACL statements if the device has enough space for a startup-config file that contains the ACLs. Enter the following command at the Global CONFIG level of the CLI.

```
BigIron RX(config)# system-max ip-filter-sys 5000
```

Syntax: [no] system-max ip-filter-sys <*num*>

Enter up to 8000 for <*num*>. The default is 4000 statements.

You can load ACLs dynamically by saving them in an external configuration file on a flash card or a 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.

ACL-based inbound mirroring

With IronWare Release 02.4.00, 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 using a protocol analyzer.

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
- Support with ACL CAM Sharing Enabled.
- 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.
- Mirror (analyzer) ports cannot be assigned to the 16x10G module. You can monitor traffic on 16x10 ports.

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.

```
BigIron RX(config)# interface ethernet 4/1
BigIron RX(config-if-e10000-4/1)# acl-mirror-port ethernet 6/1
BigIron RX(config-if-e10000-4/1)# interface ethernet 4/2
BigIron RX(config-if-e10000-4/2)# acl-mirror-port ethernet 6/2
Error: 4/2 and 4/1 should have the same ACL mirror port
```

Configuring ACL-based inbound mirroring

The following sections describe how to configure ACL-based Inbound Mirroring on a device router:

- Creating an ACL with a Mirroring Clause
- Applying the ACL to an Interface
- Specifying a Destination Mirror Port
- Specifying the Destination Mirror Port for IP Receive ACLs

Creating an ACL with a mirroring clause

The **mirror** keyword has been added for inclusion in IPv4, L2 and IPv6 ACL clauses to direct traffic that meets the clause to be sent to another port. In the following examples, the ACL is used to direct IP traffic to a mirror port.

Example of ACL-based Mirroring Supported for IPv4 ACLs.

```
BigIron RX(config)#access-list 101 permit ip any any mirror
```

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.

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.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e10000-1/1)# ip access-group 101 in
```

Specifying the destination mirror port

You can specify physical ports or a trunk 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.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 1/1 to 1/2
BigIron RX(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 trunk ports

You can mirror the traffic that has been selected by ACL-based Inbound Mirroring from a trunk by configuring a destination port within the trunk configuration as shown.

```
BigIron RX(config)# trunk switch ethernet 1/1 to 1/2
BigIron RX(config-trunk-1/1-1/2)# acl-mirror-port ethe-port-monitored 1/1
ethernet 1/3
```

Syntax: [no] acl-mirror-port ethernet-port-monitored [slot/port] ethernet [slot/port]

The [slot/port] variable specifies a port in the trunk that ACL-mirror traffic will be mirrored from.

The **ethernet** [slot/port] variable specifies port that ACL-mirror traffic from the trunk will be mirrored to.

You can also use the ACL-mirroring feature to mirror traffic from a single port within a trunk by using the **config-trunk-ind** command as shown in the following example.

```
BigIron RX(config)# trunk switch ethernet 1/1 to 1/2
BigIron RX(config-trunk-1/1-1/2)# config-trunk-ind
BigIron RX(config-trunk-1/1-1/2)# acl-mirror-port ethe-port-monitored 1/1
ethernet 1/3
```

The following considerations apply when configuring ACL-based mirroring with trunks:

- You must configure ACL-mirroring for a trunk within the trunk configuration as shown in the examples. Attempting to configure ACL-mirroring at the interface level for a port that is contained within a trunk will fail and display the following message

  ```
  Error: please use trunk config level to configure ACL based mirroring on trunk
  port.
  ```

- If an individual port is configured for ACL-based Mirroring, you cannot add it to a trunk. If you want to add it to a trunk, you must remove it from ACL-based mirroring first. Then you can add it to a trunk. It can then be configured for either ACL-based trunk mirroring or for Mirroring an individual port within a trunk.

If you attempt to add a port that is configured for ACL-based Mirroring to a port, the following message will display:

```
ACL port is configured on port 2/1, please remove it and try again.
Trunk transaction failed: Trunk Config Vetoed
```

- Deleting a trunk with ACL-based Mirroring Configured: When a trunk is deleted, the ACL-based Mirroring configuration is propagated to the individual ports that made up the trunk.

  Example: If the trunk is configured as shown.

  ```
  BigIron RX(config)# trunk switch ethernet 4/1 to 4/2
  BigIron RX(config-trunk-4/1-4/2)# acl-mirror-port ethe-port-monitored 4/1 ethe
  4/3
  ```

  And then you delete the trunk as shown.

  ```
  BigIron RX(config)# no trunk switch ethernet 4/1 to 4/2
  ```

  The configuration for ACL-based mirroring will be propagated to ports 4/1 and 4/2 as shown in the following.

  ```
  interface ethernet 4/1
     acl-mirror-port ethernet 4/3
  interface ethernet 4/2
     acl-mirror-port ethernet 4/3
  ```

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..

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.

```
BigIron RX(config)# vlan 10
BigIron RX(config-vlan-10)# tagged ethernet 4/1 to 4/3
BigIron RX(config-vlan-10)# router-interface ve 10

BigIron RX(config)# interface ethernet 4/1
BigIron RX(config-if-e10000-4/1)# acl-mirror-port ethernet 5/1
BigIron RX(config)# interface ve 10
BigIron RX(config-vif-10)# ip address 10.10.10.254/24
BigIron RX(config-vif-10)# ip access-group 102 in
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 4/3
BigIron RX(config-if-e10000-4/3)# acl-mirror-port ethernet 5/1
```

Configuring numbered and named ACLs

When you configure ACLs, you can refer to the ACL by a numeric ID or by an alphanumeric name (except for super ACLs, which must be assigned numeric IDs). The commands to configure numbered ACLs are different from the commands to configure named ACLs:

- To identify an ACL by a numeric ID, use 1 – 99 for a standard ACL, 100 – 199 for an extended ACL, and 500 – 599 for a super ACL. This document refers to these ACLs as *numbered ACLs*.
- To identify an ACL by a name, first specify whether the ACL is standard or extended, then specify the name. This document refers to these ACLs as *named ACLs*. Super ACLs must be configured with numeric IDs only.

You can configure up to 100 standard named or numbered IP ACLs, 100 extended named or numbered IP ACLs, and 100 numbered super ACLs. Regardless of how many ACLs you configure, the device can support a maximum of 1024 ACL entries, associated with the ACLs in any combination.

Configuring standard numbered ACLs

This section describes how to configure standard numbered ACLs with numeric IDs:

- For configuration information on named ACLs, refer to "Configuring standard or extended named ACLs" on page 529.
- For configuration information on extended ACLs, refer to "Configuring extended numbered ACLs" on page 520.

Standard ACLs permit or deny packets based on source IP addresses. 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 device, refer to "ACL IDs and entries" on page 513.

To configure a standard ACL and apply it to outgoing traffic on port 1/1, enter the following commands.

```
BigIron RX(config)# access-list 1 deny host 209.157.22.26 log
BigIron RX(config)# access-list 1 deny 209.157.29.12 log
BigIron RX(config)# access-list 1 deny host IPHost1 log
BigIron RX(config)# access-list 1 permit any
BigIron RX(config)# int eth 1/1
BigIron RX(config-if-e10000-1/1)# ip access-group 1 in
BigIron RX(config)# write memory
```

The commands in this example configure an ACL to deny 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*> [log]

or

Syntax: [no] access-list <*num*> deny | permit <*source-ip*>/<*mask-bits*> | <*hostname*> [log]

Syntax: [no] access-list <*num*> deny | permit host <*source-ip*> | <*hostname*> [log]

Syntax: [no] access-list <*num*> deny | permit any [log]

Syntax: [no] ip access-group <*num*> in

The 16 x 10 GE module only supports the following standard ACLs.

Syntax: [no] ip 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*>]
[match-all <*tcp-flag*s>] [match-any <*tcp-flag*s>]
[<*icmp-type*>] [established] [precedence <*name*> | <*num*>]

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.
<*destination-ip*> | <*hostname*>	Specify the destination IP address for the policy. Alternatively, you can specify the host name. If you want the policy to match on all destination 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.
log	Configures the device to generate Syslog entries and SNMP traps for packets that are denied by the access policy. If you use the **log** argument, the ACL entry is sent to the CPU for processing. Refer to "ACL logging" on page 544 for more information. You can enable logging on ACLs that support logging even when the ACLs are already in use. To do so, re-enter the ACL command and add the **log** parameter to the end of the ACL entry. The software replaces the ACL command with the new one. The new ACL, with logging enabled, takes effect immediately.

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 ACLs:

- For configuration information on named ACLs, refer to "Configuring numbered and named ACLs" on page 518.
- For configuration information on standard ACLs, refer to "Configuring standard numbered ACLs" on page 518.

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.

```
BigIron RX(config)# access-list 101 deny tcp host 209.157.22.26 any eq telnet log
BigIron RX(config)# access-list 101 permit ip any any
BigIron RX(config)# int eth 1/1
BigIron RX(config-if-e10000-1/1)# ip access-group 101 in
BigIron RX(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. Notice that some of the entries are configured to generate log entries while other entries are not thus configured.

```
BigIron RX(config)# access-list 102 perm icmp 209.157.22.0/24 209.157.21.0/24
BigIron RX(config)# access-list 102 deny igmp host rkwong 209.157.21.0/24 log
BigIron RX(config)# access-list 102 deny igrp 209.157.21.0/24 host rkwong log
BigIron RX(config)# access-list 102 deny ip host 209.157.21.100 host 209.157.22.1
log
BigIron RX(config)# access-list 102 deny ospf any any log
BigIron RX(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 device 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 device named “rkwong”.

The fourth entry denies all IP traffic from host 209.157.21.100to host 209.157.22.1 and generates Syslog entries for packets that are denied by this entry.

The fifth entry denies all OSPF traffic and generates Syslog entries for denied 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 and outgoing traffic on port 1/2 and to the incoming traffic on port 4/3.

```
BigIron RX(config)# int eth 1/2
BigIron RX(config-if-e10000-1/2)# ip access-group 102 in
BigIron RX(config-if-e10000-1/2)# exit
BigIron RX(config)# int eth 4/3
BigIron RX(config-if-e10000-4/3)# ip access-group 102 in
BigIron RX(config)# write memory
```

Here is another example of an extended ACL.

```
BigIron RX(config)# access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24
BigIron RX(config)# access-list 103 deny tcp 209.157.21.0/24 eq ftp
209.157.22.0/24
BigIron RX(config)# access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24 lt
telnet neq 5
BigIron RX(config)# access-list 103 deny udp any range 5 6 209.157.22.0/24 range
7 8
BigIron RX(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.

```
BigIron RX(config)# int eth 2/1
BigIron RX(config-if-e10000-2/1)# ip access-group 103 in
BigIron RX(config-if-e10000-2/1)# exit
BigIron RX(config)# int eth 2/2
BigIron RX(config-if-e10000-2/2)# ip access-group 103 in
BigIron RX(config)# write memory
```

Extended ACL syntax

This section presents the syntax for creating an extended 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*>]
[match-all <*tcp-flags*>] [match-any <*tcp-flags*>]
[<*icmp-type*>] [established] [precedence <*name*> | <*num*>]
[tos <*number*>] [dscp-matching <*number*>]
[802.1p-priority-matching <*number*>]
[dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>] | [dscp-marking <*number*> dscp-cos-mapping] | [dscp-cos-mapping]
[fragment] [non-fragment] [first-fragment]
[fragment-offset <*number*>]
[spi <*00000000 - ffffffff*>] [log]

Syntax: [no] access-list <*num*> deny | permit host <*ip-protocol*> any any [log]

Syntax: [no] ip access-group <*num*> in

The 16 x 10 GE module only supports the following extended ACLs.

Syntax: [no] ip 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*>]

[match-all <*tcp-flags*>] [match-any <*tcp-flags*>]
[<*icmp-type*>] [established] [precedence <*name*> | <*num*>]

General parameters for extended ACLs

The following parameters apply to any extended ACL you are creating.

<*num*>	Enter 100 – 199 for a super ACL.
deny | permit	Enter **deny** if the packets that match the policy are to be dropped; **permit** if they are to be forwarded.
any	
log	Add this parameter to the end of an ACL statement to enable the generation of SNMP traps and Syslog messages for packets denied by the ACL.You can enable logging on ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the ACL or filter command and add the log parameter to the end of the ACL or filter. The software replaces the ACL or filter command with the new one. The new ACL or filter, with logging enabled, takes effect immediately. **NOTE:** Logging must be enable on the interface to which the ACL is bound before SNMP traps and Syslog messages can be generated, even if the **log** parameter is entered. Refer to "ACL logging" on page 544.
src-mac <*src-mac*> <*mask*>	Specify the source MAC 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.
dst-mac<*dst-mac*> | <*mask*>	Specify the destination MAC 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 557. **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 557. **NOTE:** The **fragmented** and **non-fragmented** parameters cannot be used together in an ACL entry.

first-fragment	Enter this keyword if you want to filter only the first-fragmented packets. Refer to "Enabling ACL filtering of fragmented or non-fragmented packets" on page 557.
fragment-offset <*number*>	Enter this parameter if you want to filter a specific fragmented packets. Enter a value from 0 – 8191. Refer to "Enabling ACL filtering of fragmented or non-fragmented packets" on page 557.
NOTE: fragment, non-fragment, **first-fragment**, and **fragment-offset** may not be used together in the same ACL statement.	
log	Add this parameter to the end of an ACL statement to enable the generation of SNMP traps and Syslog messages for packets denied by the ACL.You can enable logging on ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the ACL or filter command and add the log parameter to the end of the ACL or filter. The software replaces the ACL or filter command with the new one. The new ACL or filter, with logging enabled, takes effect immediately. **NOTE:** Logging must be enable on the interface to which the ACL is bound before SNMP traps and Syslog messages can be generated, even if the **log** parameter is entered. Refer to "ACL logging" on page 544.

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.
match-all <*tcp-flags*> match-any <*tcp-flags*>	If you specified TCP for <*ip-protocol*>, you can specify which flags inside the TCP header need to be matched. Specify any of the following flags for <*tcp-flags*>: • + | – urg = Urgent • + | – ack= Acknowledge • + | – psh + Push • + | – rst = Reset • + | – syn = Synchronize • + | – fin = Finish Use a + or – to indicate if the matching condition requires the bit to be set to 1 (+) or 0 (–), separating each entry with a space. Enter **match-all** if you want all the flags you specified to be matched from an "established TCP session; use **match-any** of any of the flags will be matched.

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 device is running:

- any-icmp-type
- echo
- echo-reply
- information-request
- log
- 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 518.

precedence *<name>* | *<num>* The precedence option for an IP packet is set in a three-bit field following the four-bit header-length field of the packet's header. 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.

Parameter to filter packets with AHP or ESP protocols

If you entered AHP (IP Authentication Header Protocol) or ESP (Encapsulating Security Payload) for *<ip-protocol>*, then you can use the following parameter:

- *<sip>* Enables packet matching based on specific IP source addresses.

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.
- **802.1p-priority-matching** <*number*> — Only packets that have the specified 802.1p priority will be matched. Valid range is 0-7.

Parameters to alter a packet's QoS value

The parameters discussed in the sections above are used to filter packets. If the packets match the filters in an ACL statement, the packet is either permitted or denied. Once a packet is permitted, you can alter its QoS value by assigning a new DSCP value, 802.1p priority, and internal forwarding priority to the packet by doing *one* of the following:

- Specify a new QoS value to the packet by entering values for the following parameters.

Syntax: dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>

dscp-marking <*number*>	**If a packet matches the filters in the ACL statement, this parameter** assigns the DSCP value that you specify to the packet. Enter 0 – 63.
802.1p-priority-marking <*number*>	**If a packet matches the filters in the ACL statement, this parameter** assigns the 802.1p priority that you specify to the packet. Enter 0 – 7.
internal-priority-marking <*number*>	**If a packet matches the filters in the ACL statement, this parameter** assigns the internal priority that you specify to the packet. Enter 0 – 7.

For example, you enter the following.

```
dscp-marking 12 802.1p-priority-marking 1 internal-priority-marking 5
```

The packet's new QoS value is:

- 802.1p (COS) value: 1
- DSCP value: 12
- Internal Forwarding Priority: 5

- Specify a DSCP value and map that value to an internal QoS table to obtain the packet's new QoS value. Use the following parameters.

```
dscp-marking <number> dscp-cos-mapping
```

The following occurs when you use these parameters:

- Enter 0 – 63 for the **dscp-marking** <*number*> parameter.
- The **dscp-cos-mapping** parameter takes the DSCP value you specified and compares it to an internal QoS table, which is indexed by DSCP values. The corresponding 802.1p priority, internal forwarding priority, and DSCP value is assigned to the packet.

For example, if you enter `dscp-marking 7` and the internal QoS table is configured as shown in Table 97, the new QoS value for the packet is:

- 802.1p (COS) value: 7
- DSCP value: 48
- Internal Forwarding Priority: 0

TABLE 97 Example internal QOS table mappings

DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
802.1p (COS) Value	0	1	2	3	4	5	6	7	1	2	3	4	5	6	7
DSCP Value	0	15	20	3	4	5	25	48	8	9	10	11	12	13	14
Internal Forwarding Priority	7	6	5	4	3	2	1	0	1	3	0	0	0	0	0

- Use the DSCP value in the packet's header to alter its QoS value. Enter the following parameter.

  `dscp-cos-mapping`

  When you enter **dscp-cos-mapping**, the DSCP value in the packet's header is compared to a column in the internal QoS table. The 802.1p priority, internal forwarding priority, and DSCP value that are mapped to the matching column is assigned to the packet.

  For example, if the DSCP value in the packet's header is 2, using the mappings in Table 97, the packet's new QoS value is:

  - 802.1p (COS) value: 2
  - DSCP value: 15
  - Internal Forwarding Priority: 6

For more information on QoS and internal forwarding queues, refer to Chapter 18, "Configuring Quality of Service".

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*>.

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-lis**t. 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 number 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.

```
BigIron RX(config)# ip access-list standard Net1
BigIron RX(config-std-nacl)# deny host 209.157.22.26 log
BigIron RX(config-std-nacl)# deny 209.157.29.12 log
BigIron RX(config-std-nacl)# deny host IPHost1 log
BigIron RX(config-std-nacl)# exit
BigIron RX(config)# int eth 1/1
BigIron RX(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 518.

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: ip access-list standard <*string*> | <*num*>

Syntax: [no] ip access-list standard <*string*> | <*num*> deny | permit <*source-ip*> | <*hostname*> <*wildcard*> [log]

or

Syntax: [no] ip access-list standard <*string*> | <*num*> deny | permit <*source-ip*>/<*mask-bits*> | <*hostname*> [log]

Syntax: [no] ip access-list standard <*string*> | <*num*> deny | permit host <*source-ip*> | <*hostname*> [log]

Syntax: [no] ip access-list standard <*string*> | <*num*> deny | permit any [log]

Syntax: [no] ip access-group <*num*> in

The **standard** parameter indicates the ACL type.

The 16 x 10 GE module only supports the following standard named ACLs.

Syntax: [no] ip access-list standard <string> | <num> deny | permit <source-ip> | <hostname> | <source-ip>/mask-bits> | <hostname><wildcards> [log]

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 log
access-list 1 deny 209.157.22.0 0.0.0.255 log
access-list 1 permit any
access-list 101 deny tcp any any eq http log
```

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 518.

Configuration example for extended ACL

To configure a named extended ACL entry, enter commands such as the following.

```
BigIron RX(config)# ip access-list extended "block Telnet"
BigIron RX(config-ext-nacl)# deny tcp host 209.157.22.26 any eq telnet log
BigIron RX(config-ext-nacl)# permit ip any any
BigIron RX(config-ext-nacl)# exit
BigIron RX(config)# int eth 1/1
BigIron RX(config-if-e10000-1/1)# ip access-group "block Telnet" in
```

Syntax: [no] ip access-list extended <*string*> | <*num*> deny | permit <*ip-protocol*>
<*source-ip*> | <*hostname*> <*wildcard*>
[<*operator*> <*source-tcp/udp-port*>]
<*destination-ip*> | <*hostname*> <*wildcard*>
[<*operator*> <*destination-tcp/udp-port*>]
[match-all <*tcp-flags*>] [match-any <*tcp-flags*>]
[<*icmp-type*>] [established] [precedence <*name*> | <*num*>]
[tos <*number*>] [dscp-matching <*number*>]
[802.1p-priority-matching <*number*>]
[dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>]
[dscp-marking <*number*> dscp-cos-mapping]
[dscp-cos-mapping]
[fragment] [non-fragment] [first-fragment]
[fragment-offset <*number*>]
[spi <*00000000 - ffffffff*>] [log]

The 16 x 10 GE module only supports the following extended named ACLs.

Syntax: [no] ip access-list extended<*string*> | <*num*> deny | permit <*ip-protocol*>
<*source-ip*> | <*hostname*> <*wildcard*>
[<*operator*> <*source-tcp/udp-port*>]
<*destination-ip*> | <*hostname*> <*wildcard*>
[<*operator*> <*destination-tcp/udp-port*>]

[match-all <*tcp-flags*>] [match-any <*tcp-flags*>]
[<*icmp-type*>] [established] [precedence <*name*> | <*num*>]

Syntax: [no] ip access-list extended <*string*> | <*num*> deny | permit host <*ip-protocol*> any any [log]

Syntax: [no] ip access-group <*num*> in

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 520.

Configuring super ACLs

This section describes how to configure super ACLs with numeric IDs:

- For configuration information on named ACLs, refer to "Configuring standard or extended named ACLs" on page 529.
- For configuration information on extended ACLs, refer to "Configuring extended numbered ACLs" on page 520.
- Egress Super ACLs are not supported on the RX-BI=16XG (16 x 10 GE) modules

Super ACLs can match on fields in a Layer 2 or Layer 4 packet header. You can configure up to 99 super ACLs, using the number range 500 - 599. For the number of ACL entries supported on a device, refer to "ACL IDs and entries" on page 513.

Super ACL syntax is keyword-based. You specify the conditions to match as keyword-value pairs. Each keyword-value pair (called a "match-item") specifies a field in the packet header (L2, L3 or L4) to be checked, and gives the allowable value for this field. Fields not specified are called "don't care" fields, and are considered to be matched. The match-items may be specified in any order with one exception: because of its variable length, tcp-flags must be specified as the last item in a filter. The complete syntax of super ACLs is described in the next section.

NOTE
Super ACLs are not supported on management interfaces or outbound ACLs on RX-BI-16XG (16 x 10 GE) interfaces.

Super ACL filters

Some super ACL filters are shown in the following examples.

The following filter denies IPv4 TCP packets.

```
BigIron RX(config)#access-list 500 deny ip-protocol tcp
```

The following filter denies any packet with a source MAC address of 0000.0000.0011 and a source IP address from 30.30.30.0 to 30.30.30.255.

```
BigIron RX(config)#access-list 500 deny src-mac 0000.0000.0011
         ffff.ffff.ffff. sip 30.30.30.0/24
```

The following filter denies any IPv4 packet passing through the interface.

```
BigIron RX(config)#access-list 500 deny any
```

Super ACL syntax

Syntax: [no] access-list <*num*> deny | permit |
any |
log |
src-mac <*src-mac*> <*mask*> |
dst-mac <*dst-mac*> <*mask*> |
vlan-id <*vlan-id*> |
ip-pkt-len <*pkt-len*> |
ip-fragment-match {[fragment [fragment-offset <*0 - 8191*>]] | [non-fragment] | [first-fragment]} |
ip-protocol <*ip-protocol*> |
sip {<*source-ip*>/<*source-ip-mask-len*> | host <*hostname*>} |
dip {<*destination-ip*>/<*destination-ip-len*> | host <*hostname*>} |
sp <*operator*> <*source-tcp/udp-port*> |
dp <*operator*> <*destination-tcp/udp-port*> |
icmp-detail <*icmp-type-code*> |
dscp-matching <*0 – 63*> |
802.1p-priority-matching <*0 - 7*> |
ipsec-spi <*00000000 - ffffffff*> |
qos-marking {[dscp <*0 - 63*> 802.1p-priority-marking <*0 - 7*> internal-priority-marking <*0 - 7*>] |
[dscp <*0 - 63*> dscp-cos-mapping] | [use-packet-dscp dscp-cos-mapping]} | tcp-flags {[match-all <*tcp flags*>] | [match-any <*tcp flags*>] | [established]} |
<*tcp flags*> = [{+|-}urg] [{+|-}ack] [{+|-}psh] [{+|-}rst] [{+|-}syn] [{+|-}fin]
<*icmp-type-code*> = <*type*> <*code*> | <*well-known type/code*>

Most of the keywords in this syntax are self-explanatory, and work the same way as the keywords IPv4 and MAC ACLs. The QoS options are also similar to those in the IPv4 ACL, however, in super ACL the three QoS marking modes are grouped under the keyword **qos-marking** to simplify the syntax.

General parameters for super ACLs

The following parameters apply to super ACLs.

num	The ACL ID. Enter 500 – 599 for super ACLs.
deny | permit	Enter **deny** if the packets that match the policy are to be dropped; **permit** if they are to be forwarded.
any	Matches any packet
log	Enables logging for denied packets. ACL logging is disabled by default; it must be explicitly enabled on a port. **NOTE:** Logging is not currently supported on management interfaces.
src-mac	Specifies the source MAC address for the policy. Alternatively, you can specify the host name. If you want the policy to match on all source addresses, enter any.
dst-mac	Specifies the destination MAC address for the policy. Alternatively, you can specify the host name. If you want the policy to match on all destination 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.	
vlan-id	Specifies the VLAN id

ip-pkt-len <*pkt-len*>	Specifies the IP packet length to be matched.
ip-fragment-match	Enables IP fragment matching.
<*ip-protocol*>	Specifies the IP protocols to be matched.
<*sip*>	Enables packet matching based on specific IP source addresses.
<*dip*>	Enables packet matching based on specified IP destination addresses.
sp	Enables packet matching based on specified source TCP/UDP port.
dp	Enables packet matching based on specified destination TCP/UDP port.
icmp-detail	Enables packet matching based on ICMP information.
801.2-priority-matching	Enables packet matching based on the specified 802.1p priority value. Valid range is 0-7.
ipsec-spi	This parameter filters packets based on their IPSEC Security Parameter Index (SPI). Enter this value in hexadecimal. The range is 00000000 - ffffffff
qos-marking	Enables packet matching based on QoS marking.
dscp-marking	Enables packet matching based on DSCP marking.
internal-priority-marking	Enables packet matching based on internal priority marking.
tcp-flags	Enables packet matching based on TCP flags.
<*icmp-type-code*>	Enables packet matching based on ICMP type/code.

Parameters to bind super ACLs to an interface

Super ACLs can be applied to physical interfaces, trunk interfaces, and virtual interfaces. They follow the same configuration constraints as the IPv4 ACLs, for example they cannot co-exist with an IPv4 ACL on the same interface.

Syntax: [no] super-acl <*num*> in

Displaying ACL definitions

To display the ACLs configured on a device, use the **show ip access-lists** command.

Numbered ACL

For a numbered ACL, you can enter a command such as the following.

```
BigIron RX(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 number for the <*number*> parameter:

- 1 – 99 for standard ACLs
- 100 – 199 for extended ACLs
- 500 – 599 for super ACLs

Enter **all** to display all of the ACLs configured on the device.

Named ACL

For a named ACL, enter a command such as the following.

```
BigIron RX(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*>

Enter the ACL name for the <*acl-name*> parameter or the ACL number for <*acl-number*>.

Displaying of TCP/UDP numbers in ACLs

You can display the port numbers of TCP/UDP application information instead of their TCP/UDP well-known port name in the output of **show** commands and other commands that contain application port information. For example, entering the following command causes the device to display **80** (the port number) instead of http (the well-known port name).

```
BigIron(config)# ip show-acl-service-number
```

Syntax: [no] ip show-acl-service-number

By default, the device displays TCP/UDP application information in named notation.

The following table lists the ports by number and well-known name.

TABLE 98 TCP/UDP port numbers and names

Port service number	Port name	Description
1	tcpmux	TCP Port Service Multiplexer
2	compressnt-2	Management Utility
3	compressnt-3	Compression Process
5	rje	Remote Job Entry
11	systat	Active Users
13	daytime	Daytime (RFC 867)
17	qotd	Quote of the Day
18	msp	Message Send Protocol
19	chargen	Character Generator
20	ftp-data	File Transfer [Default Data]
21	ftp	File Transfer [Control]
22	ssh	SSH Remote Login Protocol
23	telnet	Telnet
25	smtp	Simple Mail Transfer
27	nsw-fe	NSW User System FE
29	msg-icp	MSG ICP
31	msg-auth	MSG Authentication

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
33	dsp	Display Support Protocol
38	rap	Route Access Protocol
39	rlp	Resource Location Protocol
41	graphics	Graphics
42	nameserver	Host Name Server
43	nicname	Who Is
44	mpm-flags	MPM FLAGS Protocol
45	mpm	Message Processing Module [recv]
46	mpm-snd	MPM [default send]
47	ni-ftp	NI FTP
48	auditd	Digital Audit Daemon
50	re-mail-ck	Remote Mail Checking Protocol
51	la-maint	IMP Logical Address Maintenance
52	xns-time	XNS Time Protocol
53	dns	Domain Name Server
54	xns-ch	XNS Clearinghouse
55	isi-gl	ISI Graphics Language
56	xns-auth	XNS Authentication
58	xns-mail	XNS Mail
61	ni-mail	NI MAIL
62	acas	ACA Services
64	covia	Communications Integrator (CI)
65	tacacs-ds	TACACS-Database Service
66	sql*net	Oracle SQL*NET
70	gopher	Gopher
71	netrjs-1	Remote Job Service
72	netrjs-2	Remote Job Service
73	netrjs-3	Remote Job Service
74	netrjs-4	Remote Job Service
76	deos	Distributed External Object Store
78	vettcp	vettcp
79	finger	Finger
80	http	World Wide Web HTTP
81	hosts2-ns	HOSTS2 Name Server
82	xfer	XFER Utility

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
83	mit-ml-dev1	MIT ML Device
84	ctf	Common Trace Facility
85	mit-ml-dev2	MIT ML Device
86	mfcobol	Micro Focus Cobol
88	kerberos	Kerberos
89	su-mit-tg	SU/MIT Telnet Gateway
90	dnsix	DNSIX Securit Attribute Token Map
91	mit-dov	MIT Dover Spooler
92	npp	Network Printing Protocol
93	dcp	Device Control Protocol
94	objcall	Tivoli Object Dispatcher
95	supdup	SUPDUP
96	dixie	DIXIE Protocol Specification
97	swift-rvf	Swift Remote Virtural File Protocol
98	tacnews	TAC News
99	metagram	Metagram Relay
100	newacct	[unauthorized use]
101	hostname	NIC Host Name Server
102	iso-tsap	ISO-TSAP Class 0
103	gppitnp	Genesis Point-to-Point Trans Net
104	acr-nema	ACR-NEMA Digital Imag. & Comm. 300
105	csnet-ns	Mailbox Name Nameserver
106	3com-tsmux	3COM-TSMUX
107	rtelnet	Remote Telnet Service
108	snagas	SNA Gateway Access Server
109	pop2	Post Office Protocol - Version 2
110	pop3	Post Office Protocol - Version 3
111	sunrpc	SUN Remote Procedure Call
112	mcidas	McIDAS Data Transmission Protocol
113	auth	Authentication Service
114	audionews	Audio News Multicast
115	sftp	Simple File Transfer Protocol
116	ansanotify	ANSA REX Notify
117	uucp-path	UUCP Path Service
118	sqlserv	SQL Services

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
119	nntp	Network News Transfer Protocol
120	cfdptkt	CFDPTKT
121	erpc	Encore Expedited Remote Pro.Call
122	smakynet	SMAKYNET
124	ansatrader	ANSA REX Trader
125	locus-map	Locus PC-Interface Net Map Ser
126	unitary	NXEdit
127	locus-con	Locus PC-Interface Conn Server
128	gss-xlicen	GSS X License Verification
129	pwdgen	Password Generator Protocol
130	cisco-fna	cisco FNATIVE
131	cisco-tna	cisco TNATIVE
132	cisco-sys	cisco SYSMAINT
133	statsrv	Statistics Service
134	ingres-net	INGRES-NET Service
135	loc-srv	DCE endpoint resolution
136	profile	PROFILE Naming System
139	netbios-ssn	NETBIOS Session Service
140	emfis-data	EMFIS Data Service
141	emfis-cntl	EMFIS Control Service
142	bl-idm	Britton-Lee IDM
143	imap4	Internet Message Access Protocol
144	news	NEWS
145	uaac	UAAC Protocol
146	iso-tp0	ISO-IP0
147	iso-ip	ISO-IP
148	cronus	CRONUS-SUPPORT
149	aed-512	AED 512 Emulation Service
150	sql-net	SQL-NET
151	hems	HEMS
152	bftp	Background File Transfer Program
153	sgmp	SGMP
154	netsc-prod	NETSC
155	netsc-dev	NETSC
156	sqlsrv	SQL Service

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
157	knet-cmp	KNET/VM Command/Message Protocol
158	pcmail-srv	PCMail Server
159	nss-routing	NSS-Routing
160	sgmp-traps	SGMP-TRAPS
163	cmip-man	CMIP/TCP Manager
164	cmip-agent	CMIP/TCP Agent
165	xns-courier	Xerox
166	s-net	Sirius Systems
167	namp	NAMP
168	rsvd	RSVD
169	send	SEND
170	print-srv	Network PostScript
171	multiplex	Network Innovations Multiplex
172	cl/1	Network Innovations CL/1
173	xyplex-mux	Xyplex
174	mailq	MAILQ
175	vmnet	VMNET
176	genrad-mux	GENRAD-MUX
177	xdmcp	X Display Manager Control Protocol
178	nextstep	NextStep Window Server
179	bgp	Border Gateway Protocol
180	ris	Intergraph
181	unify	Unify
182	audit	Unisys Audit SITP
183	ocbinder	OCBinder
184	ocserver	OCServer
185	remote-kis	Remote-KIS
186	kis	KIS Protocol
187	aci	Application Communication Interface
188	mumps	Plus Five's MUMPS
189	qft	Queued File Transport
190	gacp	Gateway Access Control Protocol
191	prospero	Prospero Directory Service
192	osu-nms	OSU Network Monitoring System
193	srmp	Spider Remote Monitoring Protocol

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
194	irc	Internet Relay Chat Protocol
195	dn6-nlm-aud	DNSIX Network Level Module Audit
196	dn6-smm-red	DNSIX Session Mgt Module Audit Redir
197	dls	Directory Location Service
198	dls-mon	Directory Location Service Monitor
199	smux	SMUX
200	src	IBM System Resource Controller
201	at-rtmp	AppleTalk Routing Maintenance
202	at-nbp	AppleTalk Name Binding
203	at-3	AppleTalk Unused
204	at-echo	AppleTalk Echo
205	at-5	AppleTalk Unused
206	at-zis	AppleTalk Zone Information
207	at-7	AppleTalk Unused
208	at-8	AppleTalk Unused
209	tam	The Quick Mail Transfer Protocol
210	z39.50	ANSI Z39.50
211	914c/g	Texas Instruments 914C/G Terminal
212	anet	ATEXSSTR
213	ipx	IPX
214	vmpwscs	VM PWSCS
215	softpc	Insignia Solutions
216	atls	Access Technology
217	dbase	dBASE Unix
218	mpp	Netix Message Posting Protocol
219	uarps	Unisys ARPs
220	imap3	Interactive Mail Access Protocol v3
221	fln-spx	Berkeley rlogind with SPX auth
222	rsh-spx	Berkeley rshd with SPX auth
223	cdc	Certificate Distribution Center
243	sur-meas	Survey Measurement
245	link	LINK
246	dsp3270	Display Systems Protocol
344	pdap	Prospero Data Access Protocol
345	pawserv	Perf Analysis Workbench

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
346	zserv	Zebra server
347	fatserv	Fatmen Server
348	csi-sgwp	Cabletron Management Protocol
371	clearcase	Clearcase
372	ulistserv	ListProcessor
373	legent-1	Legent Corporation
374	legent-2	Legent Corporation
375	hassle	Hassle
376	nip	Amiga Envoy Network Inquiry Protocol
377	tnETOS	NEC Corporation
378	dsETOS	NEC Corporation
379	is99c	TIA/EIA/IS-99 modem client
380	is99s	TIA/EIA/IS-99 modem server
381	hp-collector	hp performance data collector
382	hp-managed-node	hp performance data managed node
383	hp-alarm-mgr	hp performance data alarm manager
384	arns	A Remote Network Server System
385	ibm-app	IBM Application
386	asa	ASA Message Router Object Def.
387	aurp	Appletalk Update-Based Routing Protocol
388	unidata-ldm	Unidata LDM
389	ldap	Lightweight Directory Access Protocol
390	uis	UIS
391	synotics-relay	SynOptics SNMP Relay Port
392	synotics-broker	SynOptics Port Broker Port
393	dis	Meta5
394	embl-ndt	EMBL Nucleic Data Transfer fpkgh[pkn
395	netcp	NETscout Control Protocol
396	netware-ip	Novell Netware over IP
397	mptn	Multi Protocol Trans. Net.
398	kryptolan	Kryptolan
400	work-sol	Workstation Solutions
401	ups	Uninterruptible Power Supply
402	genie	Genie Protocol
403	decap	decap

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
404	nced	nced
405	ncld	ncld
406	imsp	Interactive Mail Support Protocol
407	timbuktu	Timbuktu
408	prm-sm	Prospero Resource Manager Sys. Man.
409	prm-nm	Prospero Resource Manager Node Man.
410	decladebug	DECLadebug Remote Debug Protocol
411	rmt	Remote MT Protocol
412	synoptics-trap	Trap Convention Port
413	smsp	Storage Management Services Protocol
414	infoseek	InfoSeek
415	bnet	BNet
416	silverplatter	Silverplatter
417	onmux	Onmux
418	hyper-g	Hyper-G
419	ariel1	Ariel 1
420	smpte	SMPTE
421	ariel2	Ariel 2
422	ariel3	Ariel 3
423	opc-job-start	IBM Operations Planning and Control Start
424	opc-job-track	IBM Operations Planning and Control Track
425	icad-el	ICAD
426	smartsdp	smartsdp
427	svrloc	Server Location
428	ocs_cmu	OCS_CMU
429	ocs_amu	OCS_AMU
430	utmpsd	UTMPSD
431	utmpcd	UTMPCD
432	iasd	IASD
433	nnsp	NNSP
435	mobilip-mn	MobilIP-MN
436	dna-cml	DNA-CML
437	comscm	comscm
438	dsfgw	dsfgw
439	dasp	dasp Thomas Obermair

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
440	sgcp	sgcp
441	decvms-sysmgt	decvms-sysmgt
442	cvc_hostd	cvc_hostd
443	ssl	http protocol over TLS/SSL
444	snpp	Simple Network Paging Protocol
445	microsoft-ds	Microsoft-DS
446	ddm-rdb	DDM-RDB
447	ddm-dfm	DDM-RFM
448	ddm-byte	DDM-BYTE
449	as-servermap	AS Server Mapper
450	tserver	COmputer Supported Telecomunication Applications
512	exec	remote process execution
513	login	remote login a la telnet
514	cmd	cmd
515	printer	spooler
518	ntalk	ntalk
519	utime	inixtime
525	timed	timeserver
526	tempo	newdate
530	courier	rpc
531	conference	chat
532	netnews	readnews
533	netwall	for emergency broadcast
539	apertus-ldp	Apertus Technologies Load Determination
540	uucp	uucpd
541	uucp-rlogin	uucp-rlogin
543	klogin	klogin
544	kshell	krcmd
550	new-rwho	new-who
554	rtsp	Real Time Stream Control Protocol
555	dsf	dfs
556	remotefs	rfs server
560	rmonitor	rmonitor
561	monitor	monitor
562	chshell	chcmd

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
564	9pfs	plan 9 file service
565	whoami	whoami
570	meter-570	demon
571	meter-571	udemon
600	ipcserver	SUN ipc sERVER
606	nqs	nqs
607	urm	urm
608	sift-uft	Sender-Initiated or Unsolicited File Transfer
609	npmp-trap	npmp-trap
610	npmp-local	npmp-local
611	npmp-gui	npmp-gui
634	ginad	ginad
666	mdqs	mdqs
667	doom	doom ID software
704	elcsd	errlog copy or server daemon
709	entrustmanager	Entrust Key Management Service Handler
729	netviewdm1	IBM Netview DM/6000 Service Handler
730	netviewdm2	IBM Netview DM/6000 send/tcp
731	netviewdm3	IBM Netview DM/6000 Server/Client
741	netgw	netrgw
742	netrcs	Network based Rev. Cont. Sys.
744	flexlm	Flexible License Manager
747	fujitsu-dev	Fujitsu License Manager
748	ris-cm	Russell Info SCI Calender Manager
749	kerberos-adm	kerberos administration
750	rfile	remote file
751	pump	pump
752	qrh	qrh
753	rrh	rrh
754	tell	send
758	nlogin	nlogin
759	con	CON
760	ns	NS
761	rxe	RXE
762	quotad	QUOTAD

TABLE 98 TCP/UDP port numbers and names (Continued)

Port service number	Port name	Description
763	cycleserv	Cycle Server
764	omserv	Om Server
765	webster	webster
767	phonebook	phone
769	vid	VID
770	cadlock-770	CADLOCK -770
771	rtip	rtip
772	cycleserv2	CYCLE Server
773	submit	SUBMIT
774	rpasswd	rpasswd
775	entomb	entomb
776	wpages	wpages
780	wpgs	wpgs
786	concert	concert
800	mdbs_daemon	mdbs_daemon
801	device	device
996	xtreelic	XTREE License Server
997	maitrd	maitrd
998	busboy	busboy
999	garcon	garcon
999	puprouter	puprouter
1000	cadlock-1000	CADILOCK - 1000
1755	mms	MMS
7070	pnm	PNM
	DECIMAL	Other well known application port number

ACL logging

In previous releases, if the logging is enabled on an interface and the log option is included on an ACL statement, packets that match a deny ACL condition are sent to the CPU for processing. Beginning with this release, a new processing method has been implemented that prevents the Syslog buffer from being overloaded with entries for every packet that has been denied. With the new method, the first packet that matches a deny ACL condition with the log option configured is sent to the CPU for logging. Then for a certain period of time, the next packets that match the deny condition are dropped in hardware; no other Syslog message is written for any denied packet during this time. Once this wait time expires, a Syslog message is written if the device receives another packet that matches the deny condition and the whole cycle is repeated.

NOTE
BigIron RX does not support permit logging.

NOTE
Logging is not currently supported on management interfaces.

Enabling the new logging method

There are no new CLI commands to enable this new processing method; it takes effect automatically if the following items have been configured:

- Syslog logging is enabled.

  ```
  BigIron RX(config)#logging on
  ```

- Add the **log** option to an ACL statement as in the following example.

  ```
  BigIron RX(config)#access-list 400 deny any any log-enabled
  ```

  or

  ```
  BigIron RX(config)#ip access-list standard hello
  BigIron RX(config-std-nacl)#deny any log
  ```

- Enable the **ip access-group enable-deny-logging** command on an interface. If this command is not enabled, packets denied by ACLs are not logged.

  ```
  BigIron RX(config)#interface ethernet 5/1
  BigIron RX(config-if-e1000-5/1)#ip access-group enable-deny-logging
  ```

Syntax: ip access-group enable-deny-logging

Specifying the wait time

You can specify how long the system waits before it sends a message in the Syslog by entering a command such as the following.

```
BigIron RX(config)# ip access-list logging-age 2
```

Syntax: ip access-list logging-age <*minutes*>

Enter 1 – 10 minutes. The default is 5 minutes.

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.

```
BigIron RX(config)#access-list 1 deny 209.157.22.0/24
BigIron RX(config)#access-list 1 permit 209.157.22.26
```

Thus, if a packet matches the first 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 device provides an alternative method that lets you upload an ACL list from a TFTP server and replace the ACLs in the device's running-config file with the uploaded list. To change an ACL, you can edit the ACL on the file server, then upload the edited ACL to the device. Then you can save the changed ACL to the device'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 device itself.

NOTE
The only valid commands that are valid in the ACL list are the **access-list** and **end** commands; other commands are ignored.

To 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 device 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.

   ```
   BigIron(config)#no access-list 1
   BigIron(config)#no access-list 101
   ```

   When you load the ACL list into the device, 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.

   ```
   BigIron(config)#access-list 1 deny host 209.157.22.26 log
   BigIron(config)#access-list 1 deny 209.157.22.0 0.0.0.255 log
   BigIron(config)#access-list 1 permit any
   BigIron(config)#access-list 101 deny tcp any any eq http log
   ```

   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 device, 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 device'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 device reads only the ACL information in the file and ignores other commands, including commands. To assign ACLs to interfaces, use the CLI.

Adding or deleting a comment

You can add or delete comments to an ACL entry.

Numbered ACLs: adding a comment

To add a comment to an ACL entry in a numbered ACL, do the following.

1. Use the **show access-list** to display the entries in an ACL. For example.

```
BigIron RX(config)# 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 the second entry in the list, enter a command such as the following.

```
BigIron RX(config)# access-list 99 remark Permit all users
```

3. Enter the filter "permit any". For example:

```
BigIron RX (config-std-nacl)# permit any
```

4. Enter a **show access-list** command displays the following:

```
BigIron RX(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. 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, no to the entire ACL.

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 529.

Numbered ACLs: deleting a comment

To delete a remark from a numbered ACL, re-enter the remark command without any remark. For example if the remarks "Permit all users" has been defined for ACL 99, remove the remark by entering the following command.

```
BigIron RX(config)# access-list 99 remark
```

Syntax: [no] access-list <*number*> remark

Note that the actual remark is blank.

Named ACLs: adding a comment to a new ACL

You can add a comment to an ACL by doing the following.

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.

```
BigIron RX(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.

```
BigIron RX(config)#ip access-list standard entry
BigIron RX(config-std-nacl)#remark Deny traffic from Marketing
BigIron RX(config-std-nacl)# deny 5.6.7.8
```

3. Enter a **show access-list** command to display the new ACL entry with its remark.

```
BigIron RX(config)#show access-list name entry
Standard IP access-list entry
deny host 1.2.4.5
permit host 5.6.7.8
ACL remark: Deny traffic from Marketing
```

Syntax: ip access-list [extended | logging-age | standard] <*acl-name*> <*acl number*> [no] remark <*string*> deny <*options*> | permit <*options*>

- **extended | logging-age | standard** parameter indicates the ACL type and logging timer setting in minutes. For more information about the logging-age parameter, refer to "ACL logging" on page 544.
- <*acl-name*> - 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").
- <*acl-num*> - ACL number (for example, super ACLs). Specify a number from 1 – 99 for standard ACLs, 100 – 199 for extended ACLs, and 500 – 599 for super ACLs.

- **remark** <*string*> - adds a comment to the ACL entry. The comment can contain up to 128 characters. Comments must be entered separately from actual ACL entries; that is, you cannot enter an ACL entry and an ACL comment with the same command. Also, in order for the remark to be displayed correctly in the output of **show** commands, a comment must be entered immediately before the ACL entry it describes.
- **deny | permit** - denies or permits specified traffic.
- <*options*> - Complete the configuration by specifying options for the standard, extended, or super ACL entry. Options you can use to configure standard or extended named ACLs are discussed in "Configuring standard or extended named ACLs" on page 529. Options for configuring super ACLs are described in "Configuring super ACLs" on page 531.

Named ACLs: deleting a comment

To delete a remark from a named ACL, enter the following command.

```
BigIron RX(config)#ip access-list standard entry
BigIron RX(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 need to create ACLs in the order you want them applied.

If you want to delete an ACL entry from within a list, enter a **show** command as discussed in "Displaying ACL definitions" on page 533 to determine the line number of the entry you want to delete. Then enter a command as shown one of the two sections below.

From numbered ACLs

If you want to delete the second entry from a numbered ACL such as ACL 99, do the following.

1. Display the contents of the list.

```
BigIron RX(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.

```
BigIron RX(config)#no access-list 99 deny host 5.6.7.8
```

3. Display the contents of the updated list.

```
BigIron RX(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-number*> parameter specifies the ACL entry to be deleted. 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, 100 – 199 for extended ACLs, or 500 – 599 for super 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 518 and "Configuring extended numbered ACLs" on page 520. Options you can use to configure super ACLs are described in "Configuring super ACLs" on page 531.

From named ACLs

To delete an ACL entry from an ACL named "entry", do the following.

1. Enter the following command to display the contents of the ACL list.

```
BigIron RX#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.

```
BigIron RX(config)#ip access-list standard entry
BigIron RX(config-std-nacl)#no deny host 10.1.1.1
```

3. Enter the **show access-list name entry** command to display the updated list.

```
BigIron RX(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: 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 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. Super ACLs must always be numbered instead of named, from 500 – 599.

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 518 show that you apply ACLs to interfaces using the **ip access-group** command. This section present additional information about applying ACLs to interfaces. Configuration examples for super ACLs appear in the section "Configuring super ACLs" on page 531.

Reapplying modified ACLs

If you make an ACL configuration change, you must reapply the ACLs to their interfaces for the change to take 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

ACL automatic rebind

Beginning wirh release 02.6.00, the ACL automatic rebind feature allows the newly changed ACL filter definitions to be automatically applied to the ports where the ACL was bound without using the "**ip rebind-acl**" command.

NOTE
Brocade recommends that this feature only be used when a small number of ACL filters are configured, otherwise a delay may be observed.

Enter commands such as the following to enable ACL automatic rebind.

```
BigIron RX(config)# auto-acl-rebind
```

Syntax: [no] auto-acl-rebind

Manually setting the ACL rebind

To reapply ACLs following an ACL configuration change, enter the following command at the global CONFIG level of the CLI.

```
BigIron RX(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 the inbound traffic direction only. The virtual interface is used for routing between VLANs, and contains all the ports within the VLAN. 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.

NOTE
Applying an ACL to a subset of physical interfaces under a virtual routing interface multiplies the amount of CAM used by the number of physical interfaces specified. An ACL that successfully functions over a whole virtual routing interface may fail if you attempt to apply it to a subset of physical interfaces.

To apply an ACL to a subset of ports within a virtual interface, enter commands such as the following.

```
BigIron RX(config)# vlan 10 name IP-subnet-vlan
BigIron RX(config-vlan-10)# untag ethernet 1/1 to 2/12
BigIron RX(config-vlan-10)# router-interface ve 1
BigIron RX(config-vlan-10)# exit
BigIron RX(config)# access-list 1 deny host 209.157.22.26 log
BigIron RX(config)# access-list 1 deny 209.157.29.12 log
BigIron RX(config)# access-list 1 deny host IPHost1 log
BigIron RX(config)# access-list 1 permit any
BigIron RX(config)# interface ve 1
BigIron RX(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*> [<*slot*>/<*portnum*>...] to <*slot*>/<*portnum*>

NOTE
The timer for logging packets denied by Layer 2 filters is separate.

Configuring the Layer 4 session log timer

You can configure the Layer 4 session log timer, which tracks packets explicitly denied by an ACL.

The first time an ACL entry denies a packet, the software immediately generates a Syslog entry and SNMP trap. The software also starts the Layer 4 session log timer. When the timer expires, the software generates a single Syslog entry for each ACL entry that has denied a packet. The message indicates the number of packets denied by the ACL entry from the time that the timer was started. If no ACL entries explicitly deny packets during an entire timer interval, the timer stops. The timer restarts when an ACL entry explicitly denies a packet.

For example, to set the timer interval to 2 minutes, enter the following command.

```
BigIron RX(config)# ip access-list logging-age 2
```

Syntax: ip access-list logging-age <*minutes*>

You can set the timer to between 1 and 10 minutes. The default is 5 minutes.

Displaying ACL log entries

The first time an entry in an ACL denies a packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for packets denied by ACLs are at the warning level of the Syslog.

When the first Syslog entry for a packet denied by an ACL is generated, the software starts an ACL timer. After this, the software sends Syslog messages every 1 to 10 minutes, depending on the value of the timer interval. If an ACL entry does not permit or deny any packets during the timer interval, the software does not generate a Syslog entry for that ACL entry.

NOTE
For an ACL entry to be eligible to generate a Syslog entry for denied packets, logging must be enabled for the entry. The Syslog contains entries only for the ACL entries that deny packets and have logging enabled.

To display Syslog entries, use one of the following methods.

Enter the following command from any CLI prompt.

```
BigIron RX(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:
Oct 13 16:24:29:N:Switch Fabric 5 temperature 59.875 C degrees is normal

Dynamic Log Buffer (50 lines):Oct 13 17:19:36:I:running-config was changed from
telnet client 192.168.9.181
Oct 13 17:06:18:I:running-config was changed from telnet client 192.168.9.181
Oct 13 16:57:44:I:ACL: entry modified from telnet session
Oct 13 16:57:40:I:ACL: entry modified from telnet session
Oct 13 16:57:32:I:ACL: entry added from telnet session
Oct 13 16:53:04:I:ACL: 10 modified from telnet session
.
.
.
```

QoS options for IP ACLs

QoS options enable you to perform QoS for packets that match the ACLs. Using an ACL to perform QoS is an alternative to the following methods.

- Directly setting the internal forwarding priority based on incoming port, VLAN membership, and so on. (This method is described in "Assigning QoS priorities to traffic" on page 469.)
- Enabling the IP ToS-based QoS feature described in "Configuring ToS-based QoS" on page 470.

  NOTE
  If you use an ACL on an interface, ToS-based QoS assumes that the ACL will perform QoS for all packets except the packets that match the **permit ip any any** ACL.

For a list of supported QoS ACL options refer to "Using ACL QoS options to filter packets" on page 527

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.

```
BigIron RX(config)# acl-duplication-check-disable
```

Syntax: [no] acl-duplication-check-disable

This command is disabled by default.

ACL accounting

The device monitors the number of times an ACL is used to filter incoming or outgoing traffic on an interface.This feature is disabled by default.

To enable ACL accounting, enter a commnad such as the folllowing.

```
BigIron RX(config)# acl-accounting-enable
```

Syntax: [no] acl-accounting-enable

Use the **no** form of this command to disable ACL accounting.

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.

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.

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.

```
device(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
    VE 1    111                473963(1s)
                               25540391(1m)
                               87014178(5m)
                               112554569(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.
* The Total In 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

Displaying statistics for an interface

To display statistics for an interface, enter commands such as the following.

```
BigIron RX(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.
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*>]

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.

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.

```
BigIron RX(config)# clear access-list all
```

Syntax: clear access-list all | ethernet <*slot*>/<*port*> | ve <*ve-num*>

Enter **all** to clear all statistics for all ACLs.

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.

Enabling ACL filtering of fragmented or non-fragmented packets

By default, when an extended ACL is applied to a port, the port will use the ACL to permit or deny the first fragment of a fragmented packet, but forward subsequent fragments of the same packet in hardware. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.

To define an extended 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

```
BigIron RX(config)# access-list 111 deny ip any any fragment
BigIron RX(config)# int eth 1/1
BigIron RX(config-if-e10000-1/1)# ip access-group 111 in
BigIron RX(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 522 for the complete syntax for extended ACLs.

Refer to "Super ACL syntax" on page 532 for the complete syntax for super ACLs.

Named ACLs

```
BigIron RX(config)# ip access-list extended entry deny ip any any fragment
BigIron RX(config)# int eth 1/1
BigIron RX(config-if-e10000-1/1)# ip access-group entry in
BigIron RX(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*>] [ip-pkt-len <*value*>] [log] [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 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. You must always specify a number (from 500 – 599) for super ACLs

Enter the **fragment** parameter to allow the ACL to filter fragmented packets. Use the **non-fragmented** parameter 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.

ACL filtering for traffic switched within a virtual routing interface

By default, a device 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 device to filter switched traffic within a virtual routing interface. When you enable the filtering, the device 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.

```
BigIron RX(config-vif-1)# ip access-group ve-traffic in
```

Syntax: [no] ip access-group ve-traffic in

ICMP filtering for extended ACLs

Extended 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.

```
BigIron RX(config)# access-list 109 deny icmp any any echo
```

or

```
BigIron RX(config)# access-list 109 deny icmp any any 8 0
```

Syntax: [no] access-list <*num*> deny | permit icmp any any [log] <*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 99 on page 559 for valid values.

Named ACLs

For example, to deny the administratively-prohibited message type in a named ACL, enter commands such as the following.

```
BigIron RX(config)# ip access-list extended entry
BigIron RX(config-ext-nacl)# deny ICMP any any administratively-prohibited
```

or

```
BigIron RX(config)# ip access-list extended entry
BigIron RX(config-ext-nacl)#deny ICMP any any 3 13
```

Syntax: [no]ip access-list extended <*acl-name*>
deny | permit host icmp any any [log] <*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 99 on page 559 for valid values.

TABLE 99 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
NOTE: 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
host-redirect	5	1
host-tos-redirect	5	3
host-tos-unreachable	3	12
host-unreachable	3	1
information-request	15	0

TABLE 99 ICMP message types and codes (Continued)

ICMP message type	Type	Code
log		
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:** This includes all redirects.	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 **NOTE:** This includes all unreachable messages	3	x

Troubleshooting ACLs

Use the following methods to troubleshoot an ACL:

- To determine whether an ACL entry is correctly matching packets, add the **log** option to the ACL entry, then reapply the ACL. This forces the device to send packets that match the ACL entry to the CPU for processing. The **log** option also generates a Syslog entry for packets that are permitted or denied by the ACL entry.
- To determine whether the issue is specific to fragmentation, remove the Layer 4 information (TCP or UDP application ports) from the ACL, then reapply the ACL.

If you are using another feature that requires ACLs, use the same ACL entries for filtering and for the other feature.

Chapter

Policy-Based Routing 22

In this chapter

Policy-Based Routing (PBR)

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 clauses in the extended ACL.

You can configure the device 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 are available, the packet is routed in the normal way.

Configuration considerations

- 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 ACLs, 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 6 instances of a route map, up to 6 ACLs in a matching policy of each route map instance, and up to 6 next hops in a set policy of each route map instance.
- ACLs with the **log** option configured should not be used for PBR purposes.

- PBR ignores explicit or 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. PBR also ignores any deny clauses in an ACL. Traffic that matches a deny clause is routed normally using Layer 3 paths.
- 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.
- PBR is not supported for fragmented packets. If the PBR's ACL filters on Layer 4 information like TCP/UDP ports, fragmented packed are routed normally.
- You can change route maps or ACL definitions dynamically and do not need to rebind the PBR policy to an interface.
- The CAM can hold up to 1024 ACL, PBR, and Rate Limiting entries and this maximum is divided as follows:
  - ACL – 416 entries
  - Rate Limiting – 416, entries shared with PBR

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.
- Configure a route map that matches on the ACLs and sets the route information.
- Apply the route map to an interface.

Configure the ACLs

PBR uses route maps to change the routing attributes in IP traffic. This section shows an example of how to configure a standard ACL to identify the source subnet for IP traffic. Refer to the Chapter 21, "Access Control List" for details on how to configure ACLs.

To configure a standard ACL to identify a source subnet, enter a command such as the following.

```
BigIron RX(config)# access-list 99 permit 209.157.23.0 0.0.0.255
```

The command in this example configures a standard ACL that permits traffic from subnet 209.157.23.0/24. After you configure a route map that matches based on this ACL, the software uses the route map to set route attributes for the traffic, thus enforcing PBR.

NOTE
Do not use an access group to apply the ACL to an interface. Instead, use a route map to apply the ACL globally or to individual interfaces for PBR, as shown in the following sections.

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

The <*num*> parameter is the access list number and can be from 1 – 99.

The deny | permit parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

The <*source-ip*> parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE
If you are configuring the ACL for use in a route map, always speciify permit. Otherwise, the BigIron RX will ignore deny clauses and packets that match deny clauses that are routed normally.

NOTE
To specify the host name instead of the IP address, the host name must be configured using the Brocade device's DNS resolver. To configure the DNS resolver name, use the ip dns server-address... command at the global CONFIG level of the CLI.

The <*wildcard*> parameter specifies the mask value to compare against the host address specified by the <*source-ip*> parameter. 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 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. To enable the software to display the CIDR masks, enter the ip show-subnet-length command at the global CONFIG level of the CLI. 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 ip access-list command.

The host <*source-ip*> | <*hostname*> parameter lets you 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.

The any parameter configures the policy to match on all host addresses.

NOTE
Do not use the log option in ACLs that will be used for PBR.

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.

```
BigIron RX(config)# route-map test-route permit 99
BigIron RX(config-routemap test-route)# match ip address 99
BigIron RX(config-routemap test-route)# set ip next-hop 192.168.2.1
BigIron RX(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] rouite-map <*map-name*> permit I 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 device, as long as system memory is available.

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 apply a PBR policy to packets that match the ACLs in a match clause. Those packets are routed normally,
- If you specify permit, the device 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 6 route map instances for comparison and ignore the rest.

Syntax: [no] match ip address <*ACL-num-or-name*>

The <*ACL-num*> parameter specifies a standard or extended ACL number or name.

Syntax: [no] set ip next hop <*ip-addr*>

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Syntax: [no] set interface null0

This command sends the traffic to the null0 interface, which is the same as dropping the 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.

```
BigIron RX(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: ip policy route-map <*map-name*>

Enabling PBR locally

To enable PBR locally, enter commands such as the following.

```
BigIron RX(config)# interface ve 1
BigIron RX(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 or virtual interfaces.

Syntax: 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 567
- “Setting the next hop” on page 568
- “Setting the output interface to the null interface” on page 569

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 1.1.1.1/24 or, if 1.1.1.x is unavailable, through 2.2.2.1/24.

```
BigIron RX(config)# access-list 101 permit tcp 10.10.10.0 0.0.0.255 eq http
5.5.5.0 0.0.0.255
BigIron RX(config)# route-map net10web permit 101
BigIron RX(config-routemap net10web)# match ip address 101
BigIron RX(config-routemap net10web)# set ip next-hop 1.1.1.1
BigIron RX(config-routemap net10web)# set ip next-hop 2.2.2.2
BigIron RX(config-routemap net10web)# exit
BigIron RX(config)# vlan 10
BigIron RX(config-vlan-10)# tagged ethernet 1/1 to 1/4

BigIron RX(config-vlan-10)# router-interface ve 1
BigIron RX(config)# interface ve 1
BigIron RX(config-vif-1)# ip policy route-map net10web
```

Syntax: [no]route-map <*map-name*> permit l deny <*num*> route-map

Syntax: [no] set ip next hop <*ip-addr*>

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Setting the next hop

The following commands configure the device 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 device 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.

```
BigIron RX(config)# access-list 50 permit 209.157.23.0 0.0.0.255
BigIron RX(config)# access-list 51 permit 209.157.24.0 0.0.0.255
BigIron RX(config)# access-list 52 permit 209.157.25.0 0.0.0.255
```

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.

```
BigIron RX(config)# route-map test-route permit 50
BigIron RX(config-routemap test-route)# match ip address 50
BigIron RX(config-routemap test-route)# set ip next-hop 192.168.2.1
BigIron RX(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.

```
BigIron RX(config)# route-map test-route permit 51
BigIron RX(config-routemap test-route)# match ip address 51
BigIron RX(config-routemap test-route)# set ip next-hop 192.168.2.2
BigIron RX(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.

```
BigIron RX(config)# route-map test-route permit 52
BigIron RX(config-routemap test-route)# match ip address 52
BigIron RX(config-routemap test-route)# set ip next-hop 192.168.2.3
BigIron RX(config-routemap test-route)# exit
```

The following command enables PBR by globally applying the test-route route map to all interfaces.

```
BigIron RX(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.

```
BigIron RX(config)# interface ve 1
BigIron RX(config-vif-1)# ip address 209.157.23.1/24
BigIron RX(config-vif-1)# ip address 209.157.24.1/24
BigIron RX(config-vif-1)# ip address 209.157.25.1/24
BigIron RX(config-vif-1)# ip policy route-map test-route
```

Setting the output interface to the null interface

The following commands configure a PBR to send all traffic from 192.168.1.204/32 to the null interface, thus dropping the traffic instead of forwarding it.

```
BigIron RX(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.

```
BigIron RX(config)# route-map file-13 permit 56
BigIron RX(config-routemap file-13)# match ip address 56
BigIron RX(config-routemap file-13)# set interface null0
BigIron RX(config-routemap file-13)# exit
```

The following command enables PBR by globally applying the route map to all interfaces.

```
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 3/11
BigIron RX(config-if-e10000-3/11)# ip address 192.168.1.204/32
BigIron RX(config-if-e10000-3/11)# ip policy route-map file-13
```

Trunk formation

When a trunk is formed, the PBR policy on the primary port applies to all the secondary ports. If a different PBR policy exists on a secondary port at a time of a trunk formation, that policy is overridden by the PBR policy on the primary port. If the primary port does not have a PBR policy, then the secondary ports will not have any PBR policy. When a trunk is removed, reload the device to restore any PBR policies that were originally configured on the secondary ports.

Chapter

23

Configuring IP Multicast Protocols

In this chapter

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 transmit 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.

device supports 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. The 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. Also, the CAM can hold up to 1535 IPv4 multicast entries.

NOTE
Each of the multicast protocols uses IGMP. IGMP is automatically enabled on an interface when you configure PIM or DVMRP on an interface and is disabled on the interface if you disable PIM or DVMRP on the interface.

The following are commonly used terms in discussing multicast-capable routers. These terms are used throughout this chapter.

Multicast terms

The following are commonly used terms in discussing multicast-capable routers. These terms are used throughout this chapter.

Node: Refers to a router or the BigIron RX.

Root Node: The node that initiates the tree building process. It is also the router that sends the multicast packets down the multicast delivery tree.

Upstream: Represents the direction from which a router receives multicast data packets. An upstream router is a node that sends multicast packets.

Downstream: Represents the direction to which a router forwards multicast data packets. A downstream router 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 router.

Intermediate Nodes: Routers that are in the path between source routers and leaf routers.

Leaf Nodes: Routers that do not have any downstream routers.

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.

NOTE
Multicast protocols can only be applied to 1 physical interface. You must create multiple VLANs with individual untagged ports and ve's under which you configure PIM.

Changing global IP multicast parameters

The sections below apply to PIM-DM, PIM-SM, and DVMRP.

Defining the maximum number of DVMRP cache entries

The DVMRP cache system parameter defines the maximum number of repeated DVMRP traffic being sent from the same source address and being received by the same destination address. To define this maximum, enter a command such as the following.

```
BigIron RX(config)# system-max dvmrp-mcache 500
```

Syntax: system-max dvmrp-mcache <*num*>

The <*num*> parameter specifies the maximum number of multicast cache entries for DVMRP. Enter a number from 128 – 2048. The default is 512.

Defining the maximum number of PIM cache entries

The PIM cache system parameter defines 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, enter a command such as the following.

```
BigIron RX(config)# system-max pim-mcache 999
```

Syntax: system-max pim-mcache <*num*>

The <*num*> parameter specifies the maximum number of multicast cache entries for PIM. Enter a number from 256 – 4096. The default is 1024.

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

- 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

To define boundaries for PIM enabled interfaces, enter a commands such as the following.

```
BigIron RX(config)#interface ve 40
BigIron RX(config-vif-40)#ip multicast-boundary MyFoundryAccessList
```

Syntax: [no] ip multicast-boundary <*acl-spec*> <*port-list*>

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 **port-list** parameter to define the member ports on which the ACL is applied. The ACL will be applied to the multicast traffic arriving in both directions.

Use the **no ip multicast boundary** command to remove the boundary on a PIM enabled interface.

NOTE
The ACL, MyFoundryAccessList can be configured using standard ACL syntax which can be found in the ACL section.

Displaying multicast boundaries

To display multicast boundary information, use the show **ip pim interface** command.

```
BigIron RX#show ip pim interface
---------+---------------+----+---+---------------------+------+-------------
Interface|Local          |Mode|Ver|  Designated Router  |TTL   | Multicast |
         |Address        |    |   |Address         Port |Thresh| Boundary  |
---------+---------------+----+---+---------------------+------+-------------+
     v10   10.1.2.1         SM  V2  Itself                    1  None
     v30  123.1.1.2         SM  V2  Itself                    1  None
     v40  124.1.1.2         SM  V2  Itself                    1  101
```

Syntax: show ip pim interface [ethernet <*slot*>/<*portnum*> | ve <*num*>]

The **ethernet** <*port-number*> parameter specifies which physical port.

Enter **ve** <*num*> for a virtual interface.

Passive Multicast Route Insertion (PMRI)

To prevent unwanted multicast traffic from being sent to the CPU, 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 running software release 02.4.00 and later. This feature does not apply to DVMRP traffic.

PMRI enables a Layer 3 switch running PIM to create an entry for a multicast route (e.g., (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 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*
- Router 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 aging processing when the traffic stops.

Configuring PMRI

PMRI is enabled by default. To disable PMRI, enter commands such as the following.

```
BigIron RX(config)#router pim
BigIron RX(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.

```
BigIron RX(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
```

Syntax: show ip pim sparse

Changing IGMP V1 and V2 parameters

IGMP allows Brocade routers to limit the multicast of IGMP packets to only those ports on the router that are identified as IP Multicast members.

The router actively sends out host queries to identify IP Multicast groups on the network

The following IGMP V1 and V2 parameters apply to PIM and DVMRP:

- **IGMP query interval** – Specifies how often the device queries an interface for group membership. Possible values are 1 – 3600. The default is 125.

- **IGMP group membership time** – Specifies how many seconds an IP Multicast group can remain on a device interface in the absence of a group report. Possible values are 1 – 7200. 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 – 10. 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.

```
BigIron RX(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 router will query an interface for group membership. Possible values are 1 – 3,600 seconds and the default value is 125 seconds.

To modify the default value for the IGMP (V1 and V2) query interval, enter the following.

```
BigIron RX(config)# ip igmp query 120
```

Syntax: ip igmp query-interval <*1-3600*>

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 1 – 7200 seconds and the default value is 260 seconds.

To define an IGMP (V1 and V2) membership time of 240 seconds, enter the following.

```
BigIron RX(config)# ip igmp group-membership-time 240
```

Syntax: ip igmp group-membership-time <*1-7200*>

Modifying IGMP (V1 and V2) maximum response time

Maximum response time defines how long the device 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 – 10. 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.

```
BigIron RX(config)# ip igmp max-response-time 8
```

Syntax: [no] ip igmp max-response-time <*num*>

The <*num*> parameter specifies the number of seconds and can be a value from 1 – 10. The default is 10.

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 Brocade 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.

```
BigIron RX(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.

```
BigIron RX(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

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 device Switches running software release 02.6.00 and later; 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 Brocade device, enter the following command.

```
BigIron RX(config)# ip igmp version 3
```

Syntax: 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.

```
BigIron RX(config)# interface eth 1/5
BigIron RX(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.

```
BigIron RX(config)# interface ve 3
BigIron RX(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.

```
BigIron RX(config)# interface ve 3
BigIron RX(config-vif-3)# ip igmp version 2
BigIron RX(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: 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

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.

```
BigIron RX(config)# interface ve 13
BigIron RX(config-vif-13)# ip igmp tracking
```

Syntax: ip igmp tracking

NOTE
IGMPv2 tracking will not operate correctly if the system is reloaded.

NOTE
IGMP tracking is not supported when an IGMPv3-configured port is in the EXCLUDE mode.

Creating a static IGMP group

To configure a physical port to be a permanent (static) member of an IGMP group, enter the following commands

```
BigIron RX(config)# interface ethernet 1/5
BigIron RX(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

```
BigIron RX(config)# interface ve 10
BigIron RX(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 10 – 3,600 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.

```
BigIron RX(config)# ip igmp query-interval 120
```

Syntax: ip igmp query-interval <*10-3600*>

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 20 – 7200 seconds and the default value is 140 seconds.

To define an IGMP membership time of 240 seconds, enter the following.

```
BigIron RX(config)# ip igmp group-membership-time 240
```

Syntax: ip igmp group-membership-time <*20-7200*>

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 – 10. 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.

```
BigIron RX(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 – 10. 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.

```
BigIron RX# show ip igmp group
Interface v18 : 1 groups
      group            phy-port static querier life mode    #_src
1     239.0.0.1        e4/20    no     yes          include 19
Interface v110 : 3 groups
      group            phy-port static querier life mode    #_src
2     239.0.0.1        e4/5     no     yes          include 10
3     239.0.0.1        e4/6     no     yes     100  exclude 13
4     224.1.10.1       e4/5     no     yes          include 1
```

To display the status of one IGMP multicast group, enter a command such as the following.

```
BigIron RX# show ip igmp group 239.0.0.1 detail
Display group 239.0.0.1 in all interfaces.
Interface v18 : 1 groups
      group           phy-port static querier life mode    #_src
1     239.0.0.1       e4/20    no     yes          include 19
    group: 239.0.0.1, include, permit 19 (source, life):
      (3.3.3.1 40) (3.3.3.2 40) (3.3.3.3 40) (3.3.3.4 40) (3.3.3.5 40)
      (3.3.3.6 40) (3.3.3.7 40) (3.3.3.8 40) (3.3.3.9 40) (3.3.3.10 40)
      (3.3.3.11 40) (3.3.3.12 40) (3.3.3.13 40) (3.3.3.14 40) (3.3.3.15 40)
      (3.3.3.16 40) (3.3.3.17 40) (3.3.3.18 40) (3.3.3.19 40)
Interface v110 : 1 groups
      group           phy-port static querier life mode    #_src
2     239.0.0.1       e4/5     no     yes          include 10
    group: 239.0.0.1, include, permit 10 (source, life):
      (2.2.3.0 80) (2.2.3.1 80) (2.2.3.2 80) (2.2.3.3 80) (2.2.3.4 80)
      (2.2.3.5 80) (2.2.3.6 80) (2.2.3.7 80) (2.2.3.8 80) (2.2.3.9 80)
```

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.

```
BigIron RX# show ip igmp group 224.1.10.1 tracking
Display group 224.1.10.1 in all interfaces with tracking enabled.
Interface v13 : 1 groups, tracking_enabled
      group           phy-port static querier life mode    #_src
1     224.1.10.1      e4/15    no     yes          include 3
    receive reports from 3 clients:
      110.110.110.7 110.110.110.8 110.110.110.9
```

Syntax: show ip igmp 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.

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 0.1:

This field	Displays
Group	The address of the multicast group
Phy-port	The physical port on which the multicast group was received.
Static	A "yes" entry in this column indicates that the multicast group was configured as a static group; "No" means it was not. Static multicast groups can be configured in IGMP V2 using the **ip igmp static** command. In IGMP V3, static sources cannot be configured in static groups.
Querier	"Yes" means that the port is a querier port; "No" means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the port.
Life	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. There is no "life" displayed in include mode.

Table 0.1:

This field	Displays
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.
#_src	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.
Group	If you requested a *detailed* report, the following information is displayed: • The multicast group address • The mode of the group • A list of sources from which traffic will be admitted (include) or denied (exclude) on the interface is listed. • The life of each source list. If you requested a *tracking* report, the clients from which reports were received are identified.

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.

```
BigIron RX# show ip igmp interface
query interval = 60, max response time= 3, group membership time=140
v5: default V2,         PIM dense, addr=1.1.1.2
  e4/12   has    0 groups, non-Querier (age=40), default V2
v18: default V2,        DVMRP, addr=2.2.2.1
  e4/20   has    0 groups, Querier, default V2
v20: configured V3,     PIM dense (port down), addr=1.1.20.1
v110: configured V3,    PIM dense, addr=110.110.110.1
  e4/6    has    2 groups, Querier, default V3
    group: 239.0.0.1, exclude, life=100, deny 13
    group: 224.1.10.1, include, permit 2
  e4/5    has    3 groups, Querier, default V3
    group: 224.2.2.2, include, permit 100
    group: 239.0.0.1, include, permit 10
    group: 224.1.10.1, include, permit 1
```

Syntax: show ip igmp interface [ve | ethernet <*number*> <*group-address*>]

Enter **ve** and its <*number*> or **ethernet** and its <*number*> to display information for a specific virtual routing interface or ethernet interface.

Entering an address for <*group-address*> displays information for a specified group on the specified interface.

The report shows the following information.

Table 0.2:

This field	Displays
Query interval	Displays how often a querier sends a general query on the interface.
Max response	The maximum number of seconds a client can wait before it replies to the query.

Table 0.2:

This field	Displays
Group membership time	The number of seconds multicast groups can be members of this group before aging out.
(details)	The following is displayed for each interface: • The ID of the interface • The IGMP version that it is running (default IGMP V2 or configured IGMP V3) • The multicast protocol it is running: DVMRP, PIM-DM, PIM-SM • Address of the multicast group on the interface • If the interface is a virtual routing interface, the physical port to which that interface belongs, the number of groups on that physical port, whether or not the port is a querier or a non-querier port, the age of the port, and other multicast information for the port are displayed.

Displaying IGMP traffic status

To display the traffic status on each virtual routing interface, enter the following command.

```
BigIron RX# 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 traffic

The report shows the following information.

Table 0.3:

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.

Table 0.3:

This field	Displays
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 statistics

To clear statistics for IGMP traffic, enter the following command.

```
BigIron RX# clear igmp traffic
```

Syntax: clear igmp traffic

This command clears all the multicast traffic information on all interfaces on the device.

IGMP V3 and source specific multicast protocols

When IGMP V3 and PIM Sparse (PIM-SM) is enabled, the source specific multicast service (SSM) becomes available. SSM simplifies PIM-SM by eliminating the RP and all protocols related to the RP.

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 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 98), enter commands such as the following.

```
PIMRouterA(config)# ip mroute 207.95.10.0 255.255.255.0 interface ethernet 2/3
distance 1
PIMRouterA(config)# ip mroute 0.0.0.0 0.0.0.0 interface ethernet 2/3 distance 1
PIMRouterA(config)# write memory
```

Syntax: ip mroute <*ip-addr*> interface ethernet <*slot*>/<*portnum*> | ve <*num*> [distance <*num*>]

Or

Syntax: 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*> 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 **distance** <*num*> parameter sets the administrative distance for the route. When comparing multiple paths for a route, the prefers the path with the lower administrative distance.

NOTE
Regardless of the administrative distances, the always prefers directly connected routes over other routes.

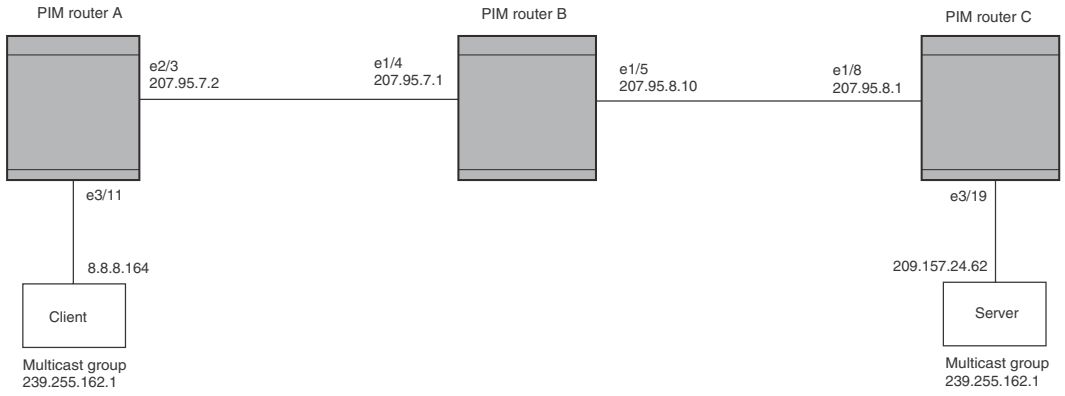
The **rpf_address** <*rpf-num*> parameter specifies an RPF number.

The example above configures two static multicast routes. The first route is for a specific source network, 207.95.10.0/24. If the 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 98 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 uses the default static route, assign a low administrative distance value. When comparing multiple paths for a route, the prefers the path with the lower administrative distance.

FIGURE 88 Example multicast static routes



To add a static route to a virtual interface, enter commands such as the following.

```
BigIron RX(config)# ip mroute 0.0.0.0 0.0.0.0 int ve 1 distance 1
BigIron RX(config)# write memory
```

Next hop validation check

Beginning with release 02.6.00, you can configure the device to perform multicast validation checks on the destination MAC address, the sender and target IP addresses, and the source MAC address.

You can enable ARP validation check on the global basis. When feature is enabled, the multicast route will only be installed when the next hop ARP has been resolved.

Configuring an ARP validation check

To enable the ARP validation check globally, enter a command such as the following.

```
BigIron RX(config)#ip mroute validate-nexthop-arp
```

Syntax: [no] ip mroute validate-nexthop-arp

Use the **no** form of the command to disable the ARP validation feature. When ARP validation is disabled, The static mroute will be installed without checking the validity of the next hop.

Enabling the next hop validate ARP timer

The **ip arp validate-nexthop-timer** command has been introduced which replaces the **ip route validate-nexthop-arp-timer** and the **ip mroute validate-nexthop-arp-timer** commands**.** The next hop validate ARP timer works only on the ARP entries created when the ARP validation check feature has been enabled. The timer is used to age out the ARP entries when the next hop goes down. All other ARP entries in the system, which are NOT created due to static routes, follow the normal ARP age timer with default value of 3 minutes.

Use the validation timer to reduce the response time where the static route with the next hop down can be replaced quickly with a route with active next hop.

To set the validation timer to 30 seconds, enter commodes such as the following.

```
BigIron RX(config)#ip arp validate-nexthop-timer 30
```

Syntax: [no] ip arp validate-nexthop-timer <*value*>

The default is 200 seconds.

The value parameter specifies the amount of time before a nexthop down is replaced by an active nexthop. Possible values are10-200 seconds.

Use the **no** form of the command to disable the validation timer.

PIM dense

NOTE
This section describes the “dense” mode of PIM, described in RFC 1075. Refer to “PIM Sparse” on page 596 for information about PIM Sparse.

NOTE
Multicast protocols can only be applied to 1 physical interface. You must create multiple VLANs with individual untagged ports and ve’s under which you configure PIM.

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 router, a network user can begin a video conference multicast from the server on R1 as shown in Figure 89. When a multicast packet is received on a PIM-capable router 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 routers. Otherwise, the multicast packet is discarded and a prune message is sent back upstream.

In Figure 89, 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. 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

As multicast packets reach these leaf routers, the routers check their IGMP databases for the group. If the group is not in a router's IGMP database, the router discards the packet and sends a prune message to the upstream router. The router 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 router 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 89 the sender with address 207.95.5.1 is sending multicast packets to the group 229.225.0.1. If a PIM router receives any groups other than that group, the router discards the group and sends a prune message to the upstream PIM router.

In Figure 90, Router R5 is a leaf node with no group members in its IGMP database. Therefore, the router must be pruned from the multicast tree. R5 sends a prune message upstream to its neighbor router R4 to remove itself from the multicast delivery tree and install a prune state, as seen in Figure 90. Router 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 89 Transmission of multicast packets from the source to host group members

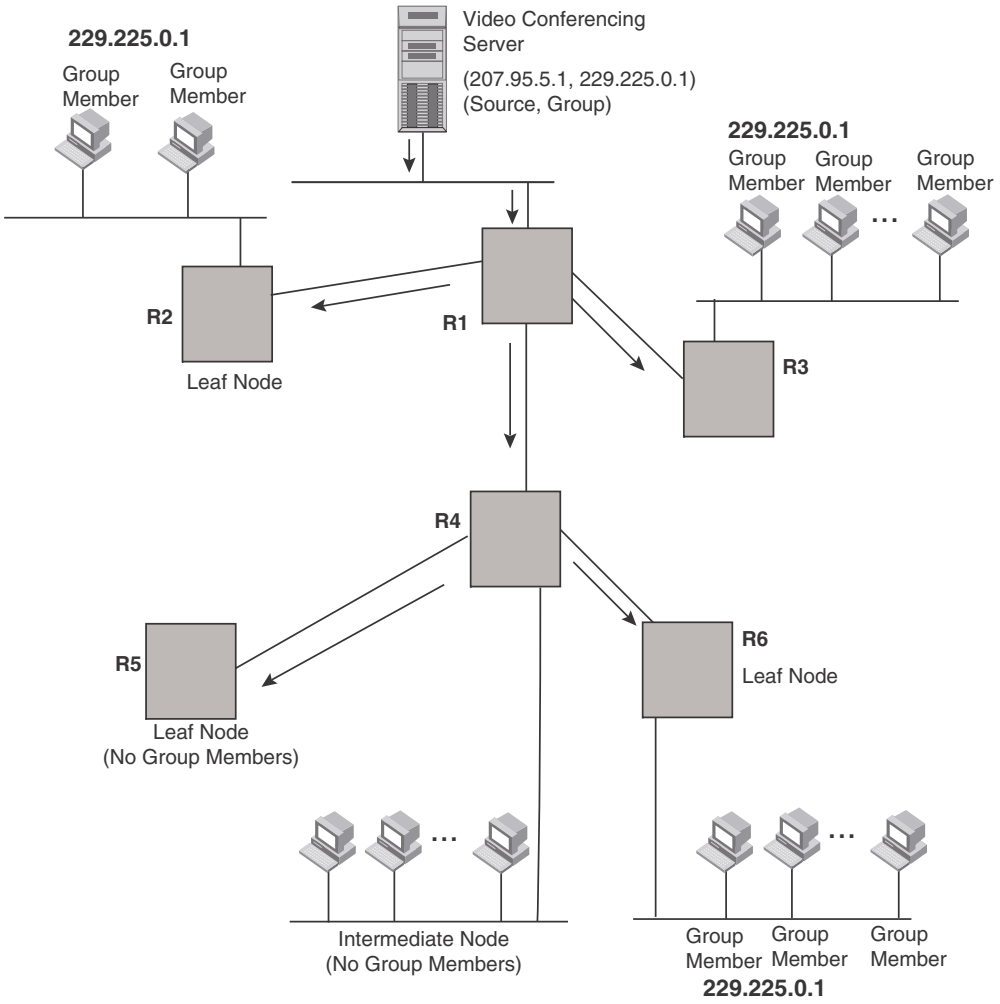
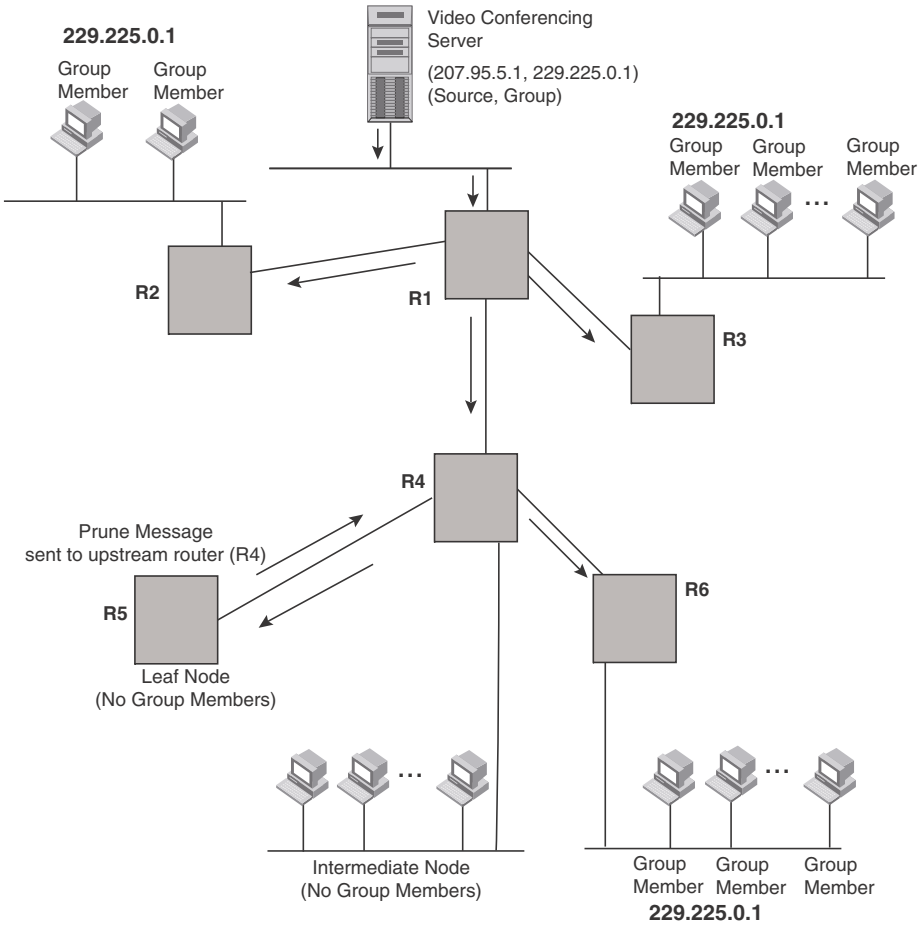


FIGURE 90 Pruning leaf nodes from a multicast tree



Grafts to a multicast tree

A PIM 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 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 598 for information about configuring PIM Sparse.

Enabling PIM on the router 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.
- Reload the software to place PIM into effect.

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 Brocade routers that connect the various buildings need to be configured to support PIM multicasts from the designated video conference server as shown in Figure 89 on page 590.

PIM is enabled on each of the Brocade routers shown in Figure 89, on which multicasts are expected. You can enable PIM on each router independently or remotely from one of the routers with a Telnet connection. Follow the same steps for each router. A reset of the router is required when PIM is first enabled. Thereafter, all changes are dynamic.

Globally enabling and disabling PIM

To globally enable PIM, enter the following command.

```
BigIron RX(config)# router pim
```

Syntax: [no] router pim

The behavior of the **[no] router pim** command was as follows:

- 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 a PIM version

To enable PIM on an interface, globally enable PIM, then enable PIM on interface 1/3, enter the following commands.

```
BigIron RX(config)# router pim
BigIron RX(config)# int e 1/3
BigIron RX(config-if-e10000-1/3)# ip address 207.95.5.1/24
BigIron RX(config-if-e10000-1/3)# ip pim
BigIron RX(config-if-e10000-1/3)# write memory
BigIron RX(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.

```
BigIron RX(config-if-e10000-1/1)# ip pim version 2
BigIron RX(config-if-e10000-1/1)# no ip pim version 1
```

To disable PIM DM on the interface, enter the following command.

```
BigIron RX(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 you need to:

- 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 router will consider a neighbor to be absent. Absence of PIM hello messages from a neighboring router indicates that a neighbor is not present.

The default value is 180 seconds.

To apply a PIM neighbor timeout value of 360 seconds to all ports on the router operating with PIM, enter the following.

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# nbr-timeout 360
```

Syntax: nbr-timeout <*60-8000*>

The default is 180 seconds.

Modifying hello timer

This parameter defines the interval at which periodic hellos are sent out PIM interfaces. Routers use hello messages to inform neighboring routers of their presence. The default rate is 60 seconds.

To apply a PIM hello timer of 120 seconds to all ports on the router operating with PIM, enter the following.

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# hello-timer 120
```

Syntax: hello-timer <*10-3600*>

The default is 60 seconds.

Modifying prune timer

This parameter defines how long a Brocade PIM router will maintain a prune state for a forwarding entry.

The first received multicast interface is forwarded to all other PIM interfaces on the router. 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.

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# prune-timer 90
```

Syntax: prune-timer <*10-3600*>

The default is 180 seconds.

Modifying the prune wait timer

The **prune-wait** command allows you to configure 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 zero to three 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 router 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 during time or in less than three seconds.

To set the prune wait time to zero, enter the following commands.

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# prune-wait 0
```

Syntax: prune-wait <*time*>

Where <*time*> has a minimum value of 60 seconds. The default is 60 seconds.

Viewing the prune wait time

To view the prune wait time, enter the following command at any level of the CLI.

```
BigIron RX(config)#show ip pim dense
Global PIM Dense Mode Settings
Hello interval: 60, Neighbor timeout: 180
Graft Retransmit interval: 180, Inactivity interval: 180
Route Expire interval: 200, Route Discard interval: 340
Prune age: 180, Prune wait: 3
```

Syntax: show ip pim dense

Modifying graft retransmit timer

The Graft Retransmit Timer defines the interval between the transmission of graft messages.

A graft message is sent by a router to cancel a prune state. When a router receives a graft message, the router responds with a Graft Ack (acknowledge) message. If this Graft Ack message is lost, the router 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.

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# graft-retransmit-timer 90
```

Syntax: graft-retransmit-timer <*60-3600*>

The default is 180 seconds.

Modifying inactivity timer

The router 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 router deletes it.

To apply a PIM inactivity timer of 90 seconds to all PIM interfaces, enter the following.

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# inactivity-timer 90
```

Syntax: inactivity-timer <*10-3600*>

The default is 180 seconds.

Selection of shortest path back to source

By default, when a multicast packet is received on a PIM-capable router 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 table below, the first four routes have the same cost back to the source. However, 137.80.127.3 will be chosen as the path to the source since it is the first one on the list. The router rejects traffic from any port other than Port V11 on which 137.80.127.3 resides.

```
    Total number of IP routes: 19
    B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
            Destination     NetMask         Gateway         Port       Cost
Type
        ..
    9       172.17.41.4     255.255.255.252*137.80.127.3    v11        2
O
            172.17.41.4     255.255.255.252 137.80.126.3    v10        2
O
            172.17.41.4     255.255.255.252 137.80.129.1    v13        2
O
            172.17.41.4     255.255.255.252 137.80.128.3    v12        2
O
    10      172.17.41.8     255.255.255.252 0.0.0.0         1/2        1
D
```

Failover time in a multi-path topology

Previously, when a port in a multi-path topology fails, multicast routers, 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 router.

No configuration is required for this feature.

Modifying the TTL

The TTL defines the minimum value required in a packet for it to be forwarded out of 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 will be forwarded. Likewise, if an interface is configured with a TTL Threshold value of 1, all packets received on that interface will be forwarded. Possible TTL values are 1 to 64. The default TTL value is 1.

To configure a TTL of 45, enter the following.

```
BigIron RX(config-if-e10000-3/24)# ip pim ttl 45
```

Syntax: ip pim ttl *<1-64>*

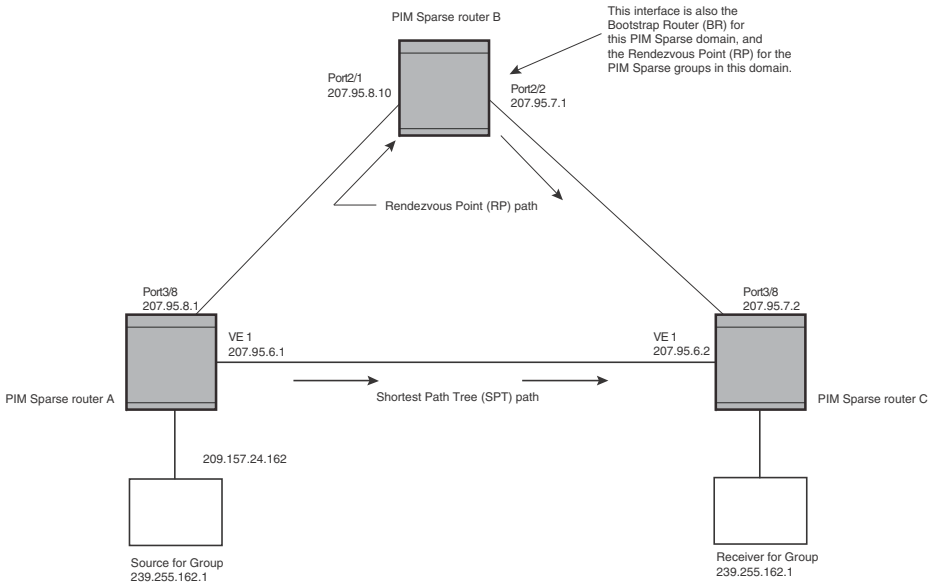
PIM Sparse

The device supports Protocol Independent Multicast (PIM) Sparse version 2. PIM Sparse provides multicasting that is especially suitable for widely distributed multicast environments. The Brocade implementation is based on RFC 2362.

In a PIM Sparse network, a 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).

PIM Sparse routers are organized into domains. A PIM Sparse domain is a contiguous set of routers that all implement PIM and are configured to operate within a common boundary. Figure 91 shows a simple example of a PIM Sparse domain. This example shows three device devices configured as PIM Sparse routers. The configuration is described in detail following the figure.

FIGURE 91 Example 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:

- **PMBR** – A PIM router that has some interfaces within the PIM domain and other interface outside the PIM domain. PBMRs connect the PIM domain to the Internet.

  NOTE
  You cannot configure a Brocade routing interface as a PMBR interface for PIM Sparse in the current software release.

- **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 91, PIM Sparse router B is the BSR. Port 2/2 is configured as a candidate BSR.
- **RP** – The RP is the rendezvous 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 91, PIM Sparse router B is the RP. Port 2/2 is configured as a candidate

Rendezvous Point (RP).

To enhance overall network performance, device use 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
Brocade recommends that you configure the same ports as candidate BSRs and RPs.

RP paths and SPT paths

Figure 91 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 router A and the recipient is attached to PIM Sparse router C. PIM Sparse router 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 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 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 forward 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 91, the device A forwards the first packet from group 239.255.162.1's 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.

NOTE
Brocade recommends that you configure the same ports as candidate BSRs and RPs

Configuring PIM Sparse

To configure a device 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.

NOTE
You cannot configure a Brocade routing interface as a PMBR interface for PIM Sparse in the current software release.

- Configure the following PIM Sparse global parameters:
  - 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
Brocade recommends that you configure the same device 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

NOTE
When PIM routing is enabled on a device, 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.

To configure basic global PIM Sparse parameters, enter commands such as the following on each device within the PIM Sparse domain.

```
BigIron RX(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 device as a candidate PIM Sparse Bootstrap Router (BSR) and candidate Rendezvous Point (RP). You can configure adevice as a PIM Sparse router without configuring the device as a candidate BSR and RP. However, if you do configure the device as one of these, Brocade recommends that you configure the device as both of these. Refer to "Configuring BSRs" on page 600.

Entering a **[no] router pim** command does the following:

- Disables PIM or DVMRP.
- Removes all configuration for PIM multicast on a device (**router pim** level) only.

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.

```
BigIron RX(config)# interface ethernet 2/2
BigIron RX(config-if-e10000-2/2)# ip address 207.95.7.1 255.255.255.0
BigIron RX(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.

```
BigIron RX(config-if-e10000-2/2)# ip pim border
```

Syntax: [no] ip pim border

NOTE
You cannot configure a Brocade routing interface as a PMBR interface for PIM Sparse in the current software release.

Configuring BSRs

In addition to the global and interface parameters in the sections above, 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 Brocade recommends that you configure the same interface on the same device as both a BSR and an RP.

This section presents how to configure BSRs. Refer to "Configuring RPs" on page 601 for instructions on how to configure RPs.

To configure the device as a candidate BSR, enter commands such as the following.

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# bsr-candidate ethernet 2/2 30 255
BSR address: 207.95.7.1, hash mask length: 30, priority: 255
```

This command configures 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*> | ve <*num*> <*hash-mask-length*> [<*priority*>]

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 **ve** <*num*> for a virtual interface.
- Enter **loopback** <*num*> for a loopback interface.

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.

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.

```
BigIron RX(config-pim-router)# rp-candidate ethernet 2/2
```

Syntax: [no] rp-candidate ethernet <*slot*>/<*portnum*> | loopback <*num*> | ve <*num*>

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 RP.

- Enter **ethernet** <*slot*>/<*portnum*> for a physical interface (port).
- Enter **ve** <*num*> for a virtual interface.
- Enter **loopback** <*num*> for a loopback interface.

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.

```
BigIron RX(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 change the group numbers for which the device is a candidate RP by deleting address ranges. For example, to delete all addresses from 224.126.22.0 – 224.126.22.255, enter the following command.

```
BigIron RX(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.

If you enter both commands shown in the example above, the net effect is that the device becomes a candidate RP for groups 224.126.0.0 – 224.126.21.255 and groups 224.126.23.0 – 224.126.255.255.

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.

```
BigIron RX(config)# clear pim rp-map
```

Syntax: clear pim rp-map

Statically specifying the RP

Brocade recommends 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 its IP 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 routers within the PIM Sparse domain. Make sure the router 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.

```
BigIron RX(config)# router pim
BigIron RX(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 the example above identifies the router interface at IP address 207.95.7.1 as the RP for the PIM Sparse domain. The device will use the specified RP and ignore group-to-RP mappings received from the BSR.

ACL based RP assignment

In patch release 02.4.00c of the device, the **rp-address** command has been enhanced to allow 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 will be 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 32 static RPs in the systems.

Configuring an ACL based RP assignment

To configure an ACL based RP assignment; enter commands such as the following.

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# rp-address 130.1.1.1 acl1
```

Syntax: rp-address <*ip_address*> [<*acl_name_or_id*>]

Use the **ip address** parameter to specify the IP address of the router you want to designate as an RP router.

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.

Displaying the static RP

Use the **show ip pim rp-set** command to display static RP and the associated group ranges.

```
BigIron RX(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.

```
BigIron RX(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
```

Anycast RP

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 device Router but re-purposes it to provide the benefits desired as described in RFC 3446.

Configuring anycast RP

To configure Anycast RP, you must do the following:

- 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

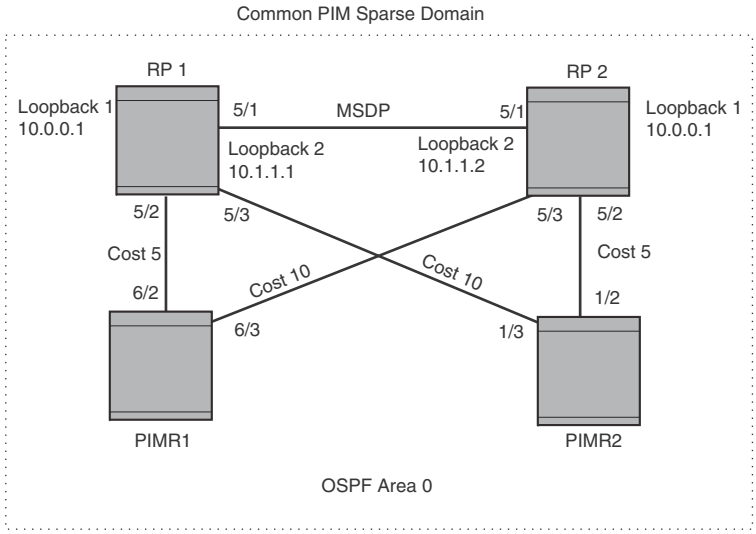
The example shown in Figure 92 is a simple Anycast-enabled network with two RPs and two PIM-SM routers. Loopback 1 in RP 1 and RP 2 have the same IP address. Loopback 2 in RP1 and Loopback 2 in RP2 have different IP addresses and are configured as MSDP peering IP addresses in a mesh configuration.

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 92 Example of an anycast RP BigIron RX



RP 1 configuration

The following commands provide the configuration for the RP 1 router in Figure 92.

```
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 92.

```
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 92.

```
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 92.

```
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
```

Route selection precedence for multicast

In patch 02.4.00c, the **route-precedence** command allows the user to specify a precedence table that dictates how routes are selected for multicast.

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 this command you may 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.

```
BigIron RX(config)# router pim
BigIron RX(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.

```
BigIron RX(config)# router pim
BigIron RX(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** parameter 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. The example below displays the default route precedence selection.

```
BigIron RX(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
```

This example displays the route precedence selection as multicast non-default, then unicast non-default, then multicast default, and then unicast default.

```
BigIron RX(config-pim-router)#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 uc-non-default mc-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
```

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 router 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 is 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, the device 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's 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.

To change the number of packets the device sends using the RP before switching to the SPT, enter commands such as the following.

```
BigIron RX(config)# router pim
BigIron RX(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.

Changing the PIM join and prune message interval

By default, the device sends PIM Sparse Join/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/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/Prune interval, enter commands such as the following.

```
BigIron RX(config)# router pim
BigIron RX(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.

MLL optimization

MLL optimization is enabled by default, except for trunks in order that trunk load sharing remains unaffected. The **ip multicast-routing optimization oif-list trunks** command can be used to turn on optimization for trunks such that the degree of 'even balance' maybe less than when not optimized.

```
BigIron RX(config)# ip multicast-routing optimization oif-list trunks
```

Syntax: ip multicast-routing optimization oif-list trunks

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

Displaying basic PIM Sparse configuration information

To display PIM Sparse configuration information, enter the following command at any CLI level.

```
BigIron RX(config-pim-router)# show ip pim sparse

Global PIM Sparse Mode Settings
  Hello interval: 60, Neighbor timeout: 180
  Bootstrap Msg interval: 130, Candidate-RP Advertisement interval: 60
  Join/Prune interval: 60, SPT Threshold: 1

Interface Ethernet e3/8
TTL Threshold: 1, Enabled
Local Address: 207.95.8.1

Interface Ve 1
TTL Threshold: 1, Enabled
Local Address: 207.95.6.1
```

Syntax: show ip pim sparse

This example shows the PIM Sparse configuration information on PIM Sparse router A in Figure 91.

This display shows the following information.

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 show the number of seconds between hello messages. 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 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's 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/Prune interval	How frequently the device sends PIM Sparse Join/Prune messages for the multicast groups it is forwarding. This field show the number of seconds between Join/Prune messages. The device sends Join/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/Prune messages for the source's group. You can change the Join/Prune interval if needed. Refer to "Changing the PIM join and prune message interval" on page 609.

This field...	Displays...
SPT Threshold	The number of packets the device sends using the path through the RP before switching to using the SPT path.
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
Local Address	Indicates the IP address configured on the port or virtual interface.

Displaying a list of multicast groups

To display PIM Sparse configuration information, enter the following command at any CLI level.

```
BigIron RX(config-pim-router)# show ip pim group

Total number of Groups: 2
Index 1         Group 239.255.162.1       Ports e3/11
```

Syntax: show ip pim group

This display shows the following information.

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.

```
BigIron RX(config-pim-router)# 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 show information displayed on a device that has been elected as the BSR. The following 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.

```
BigIron RX(config-pim-router)# show ip pim bsr

PIMv2 Bootstrap information
 BSR address  = 207.95.7.1
 BSR priority = 5
```

Syntax: show ip pim bsr

This display shows the following information.

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	**NOTE:** 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 PR advertisement message. **NOTE:** This field appears only if this device is a candidate BSR.

This field...	Displays...
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.

```
BigIron RX(config-pim-router)# 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 following example shows the message displayed on a device that is not a candidate RP.

```
BigIron RX(config-pim-router)# show ip pim rp-candidate
```

This system is not a Candidate-RP.

Syntax: show ip pim rp-candidate

This display shows the following information.

This field...	Displays...
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 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.

```
BigIron RX(config-pim-router)# 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 rp-map

This display shows the following information.

Table 0.4:

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.

```
BigIron RX(config-pim-router)# show ip pim rp-hash 239.255.162.1

  RP: 207.95.7.1, v2
    Info source: 207.95.7.1, through bootstrap
```

Syntax: show ip pim rp-hash <*group-addr*>

The <*group-addr*> parameter is the address of a PIM Sparse IP multicast group.

This display shows the following information.

Table 0.5:

This field...	Displays...
RP	Indicates the IP address of the Rendezvous Point (RP) for the specified PIM Sparse group. Following the IP address is 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. Following the IP 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 following command at any CLI level.

```
BigIron RX(config)#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 rp-set

This display shows the following information.

Table 0.6:

This field...	Displays...
Number of group prefixes	The number f 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/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 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 the device's PIM neighbors, enter the following command at any CLI level.

```
BigIron RX(config-pim-router)# show ip pim nbr

Port   Neighbor        Holdtime   Age     UpTime
                       sec        sec     sec
e3/8   207.95.8.10     180        60      900
Port   Neighbor        Holdtime   Age     UpTime
                       sec        sec     sec
v1     207.95.6.2      180        60      900
```

Syntax: show ip pim nbr

This display shows the following information.

Table 0.7:

This field...	Displays...
Port	The interface through which the device is connected to the neighbor.
Neighbor	The IP interface of the 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.
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.

Displaying information about an upstream neighbor device

You can view information about the upstream neighbor device for a given source IP address for IP PIM and DVMRP packets. For PIM, the software uses the IP route table or multicast route table to lookup the upstream neighbor device. For DVMRP, the software uses the DVMRP route table to locate the upstream neighbor device.

Enter the following command at the Privileged EXEC level of the CLI.

```
BigIron RX# show ip pim rpf 1.1.20.2
directly connected or through an L2 neighbor
```

NOTE
If there are multiple equal cost paths to the source, the **show ip pim rpf** command output may not be accurate. If your system has multiple equal cost paths, use the command **show ip pim mcache** to view information about the upstream neighbor.

The following example outputs show other messages that the device displays with this command.

```
BigIron RX# show ip pim rpf 1.2.3.4
no route
BigIron RX# show ip pim rpf 1.10.10.24
upstream neighbor=1.1.20.1 on v21 using ip route
```

Syntax: show ip pim | dvmrp rpf <*IP address*>

Where <*IP address*> is a valid source IP address

Displaying the PIM multicast cache

To display the PIM multicast cache, enter the following command at any CLI level.

```
BigIron RX(config-pim-router)# 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 mcache

This display shows the following information.

Table 0.8:

This field...	Displays...
(<*source*>, <*group*>)	The comma-separated values in parentheses is a source-group pair. The <*source*> is the PIM source for the multicast <*group*>. For example, the following entry means source 209.157.24.162 for group 239.255.162.1: (209.157.24.162,239.255.162.1) If the <*source*> value is * (asterisk), this cache entry uses the RP path. The * value means “all sources”. If the <*source*> is a specific source address, this cache entry uses the SPT path.
RP<*ip-addr*>	Indicates the RP for the group for this cache entry. **NOTE:** The RP address appears only if the RPT flag is set to 1 and the SPT flag is set to 0 (see below).
forward port	The port through which the device reaches the source.
Count	The number of packets forwarded using this cache entry.
Sparse Mode	Indicates whether the cache entry is for regular PIM (dense mode) or PIM Sparse. This flag can have one of the following values: • 0 – The entry is not for PIM Sparse (and is therefore for the dense mode of PIM). • 1– The entry is for PIM Sparse.
RPT	Indicates whether the cache entry uses the RP path or the SPT path. The RPT flag can have one of the following values: • 0 – The SPT path is used instead of the RP path. • 1– The RP path is used instead of the SPT path. **NOTE:** The values of the RP and SPT flags are always opposite (one is set to 0 and the other is set to 1).
SPT	Indicates whether the cache entry uses the RP path or the SPT path. The SP flag can have one of the following values: • 0 – The RP path is used instead of the SPT path. • 1– The SPT path is used instead of the RP path. **NOTE:** The values of the RP and SPT flags are always opposite (one is set to 0 and the other is set to 1).
Register Suppress	Indicates whether the Register Suppress timer is running. This field can have one of the following values: • 0 – The timer is not running. • 1 – The timer is running.
member ports	Indicates the device physical ports to which the receivers for the source and group are attached. The receivers can be directly attached or indirectly attached through other PIM Sparse routers.
virtual ports	Indicates the virtual interfaces to which the receivers for the source and group are attached. The receivers can be directly attached or indirectly attached through other PIM Sparse routers.
prune ports	Indicates the physical ports on which the device has received a prune notification (in a Join/Prune message) to remove the receiver from the list of recipients for the group.
virtual prune ports	Indicates the virtual interfaces ports on which the device has received a prune notification (in a Join/Prune message) to remove the receiver from the list of recipients for the group.

Displaying PIM traffic statistics

To display PIM traffic statistics, enter the following command at any CLI level.

```
BigIron RX(config-pim-router)# 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 traffic

NOTE
If you have configured interfaces for standard PIM (dense mode) on the device, statistics for these interfaces are listed first by the display.

This display shows the following information.

Table 0.9:

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/P	The number of Join/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/Xmit	The total number of IGMP messages sent and received by the device.
Total Discard/chksum	The total number of IGMP messages discarded, including a separate counter for those that failed the checksum comparison.

PIM-SSMv4

Source Specific mode is similar to Sparse mode, but instead of simply joining a group, the receiver uses IGMPv3 messages to join a group sent from a specific source. This way, instead of the receiver getting multiple copies of the same stream from multiple sources, it will only receive traffic from the one source to which it joins.

The amount of unwanted traffic in the network is reduced, but because each multicast group is associated with a particular host, different hosts can be assigned the same multicast address for different streams. This greatly increases the number of multicast groups that can be used in the network. Another added benefit of SSM is that it increases security by reducing the possibility of a rogue source disrupting the traffic from a legitimate source.

SSM defines a Short Path Tree (SPT) for multicast traffic. If SSM is enabled, the SPT is identified by an (S,G) pair, where S is a source address and G is an SSM destination address. If the SSM protocol is not enabled and before the SPT switchover, the multicast switch creates one (*, G) entry for the entire multicast group, which can have many sources. The SSM SPT is on a per-source basis and allows a client to receive multicast traffic directly from the source; that is, all joins and leaves are source-specific. You are also able to configure a specific SSM path.

SSM is limited to multicast group addresses in the 224.0.1.0 through 239.255.255.255 address range. If PIM Sparse is used as the multicast protocol, the SSM protocol should be enabled if you want to filter unwanted traffic before the Shortest Path Tree protocol switchover occurs for groups in the 232/8 range. Not configuring the SSM protocol in PIM Sparse may cause the switch or router to leak unwanted packets with the same group, but containing undesired sources, to clients. After SPT switch over, the leak stops and source specific multicast works correctly even without configuring the SSM protocol.

If the SSM protocol is enabled, one (S,G) entry is created for every source of the multicast group, even for sources with non-existent traffic. For example, if there are 1,000 sources in the group, 1,000 (S,G) entries will be created. Therefore, enabling the SSM protocol for PIM-SM requires more software resources than leaving the protocol disabled.

Enabling SSM

To enable the SSM protocol, IGMP v3 and PIM-SM must be enabled. Enter the **ssm-enable** command under the router pim level to globally enable the SSM protocol on the device.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# ssm-enable
```

Syntax: [no] ssm-enable [range <*ip-address-prefix*/<*mask-length*>]

Enter the IP address range <*ip-address-prefix* / <*mask-length*>. The default is 232/8.

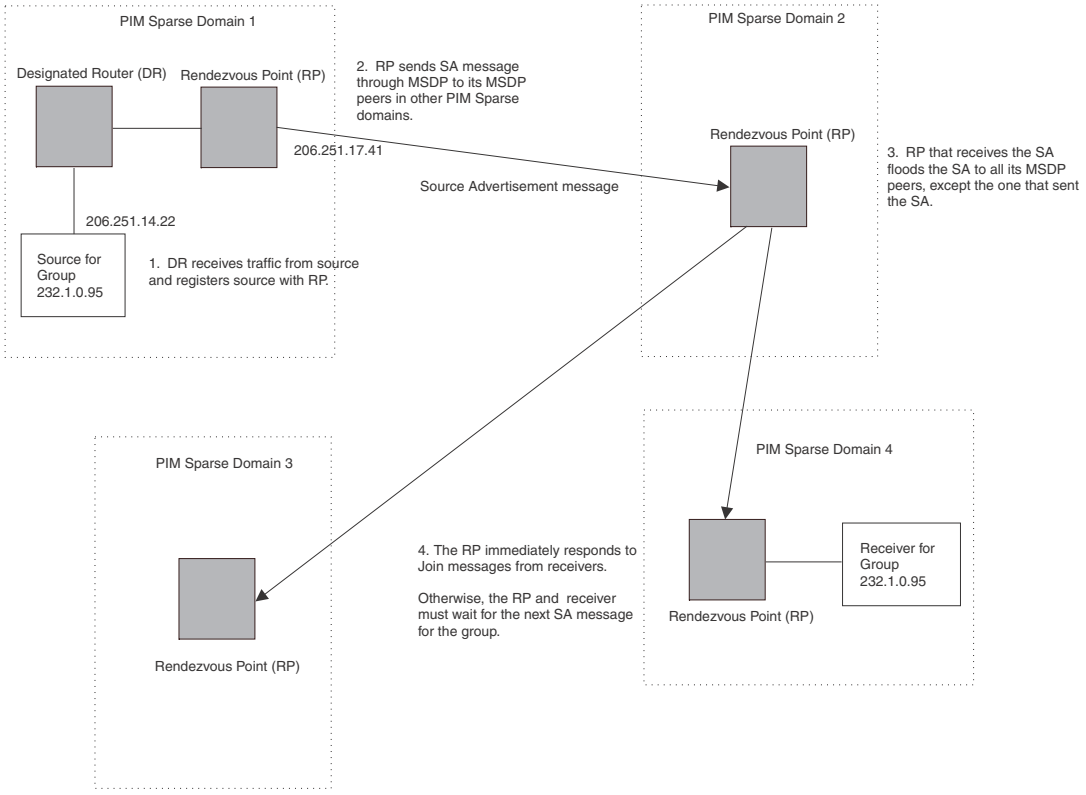
Configuring Multicast Source Discovery Protocol (MSDP)

The Multicast Source Discovery Protocol (MSDP) is used by Protocol Independent Multicast (PIM) Sparse routers to exchange routing information for PIM Sparse multicast groups across PIM Sparse domains. Routers running MSDP can discover PIM Sparse sources that are in other PIM Sparse domains.

PIM Sparse routers use MSDP to register PIM Sparse multicast sources in a domain with the Rendezvous Point (RP) for that domain.

Figure 93 shows an example of some PIM Sparse domains. For simplicity, this example show only one Designated Router (DR), one group source, and one receiver for the group. Only one PIM Sparse router within each domain needs to run MSDP.

FIGURE 93 PIM Sparse domains joined by MSDP routers



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 domain's RP. 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 of its peers by sending 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 interface with its peer. By default, the IP address included in the RP address field of the SA message is the IP address of the originating RP, but an SA message can use the IP address of any interface on the originating RP. (The interface is usually a loopback interface.)

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 93 shows only one peer for the MSDP router (which is also the RP here) in domain 1, so the Source Active message goes to only that peer. When an MSDP router has multiple peers, it sends a Source Active message to each of those peers. Each peer sends the Source Advertisement to its other MSDP peers. The RP that receives the Source Active message also sends a Join message for the group if the RP that received the message has receivers for the group.

Peer Reverse Path Forwarding (RPF) flooding

When the MSDP router (also the RP) in domain 2 receives the Source Active message from its peer in domain 1, the MSDP router in domain 2 forwards the message to all its other peers. The propagation process is sometimes called “peer Reverse Path Forwarding (RPF) flooding”. This term refers to the fact that the MSDP router uses its PIM Sparse RPF tree to send the message to its peers within the tree. In Figure 93, the MSDP router floods the Source Active message it receives from its peer in domain 1 to its other peers, in domains 3 and 4.

Note that the MSDP router in domain 2 does not forward the Source Active back to its peer in domain 1, because that is the peer from which the router received the message. An MSDP router 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 router 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 interdomain operations.

The MSDP routers in domains 3 and 4 also forward the Source Active message to all their peers except the ones that sent them the message. Figure 93 does not show additional peers.

Source active caching

When an MSDP router that is also an RP receives a Source Active message, the RP checks its 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 DR sends a Join message for that receiver back to the DR in the domain from which the Source Active message came. Usually, the DR is also the MSDP router that sent the Source Active message.

In Figure 93, if the MSDP router 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 RP for the source, in this case the RP in domain 1.

Some MSDP routers that are also RPs can cache Source Active messages. If the RP is not caching Source Active messages, the RP does not send a Join message unless it already has a receiver that wants to join the group. Otherwise, the RP does not send a Join message and does not remember the information in the Source Active message after forwarding it. If the RP receives a request from a receiver for the group, the RP and receiver must wait for the next Source Active message for that group before the RP can send a Join message for the receiver.

However, if Source Active caching is enabled on the MSDP and RP router, the RP caches the Source Active messages it receives. In this case, even if the RP does not have a receiver for a group when the RP receives the Source Active message for the group, the RP can immediately send a Join for a new receiver that wants to join the group, without waiting for the next Source Active message from the RP in the source’s domain.

The size of the cache used to store MSDP Source Active messages is 8K

Configuring MSDP

To configure MSDP on a device, perform the following tasks:

- Enable MSDP
- Configure the MSDP peers

NOTE
The PIM Sparse Rendezvous Point (RP) is also an MSDP peer.

Routers that run MSDP must also run BGP. Also, the source address used by the MSDP router must be the same source address used by BGP.

Enabling MSDP

NOTE
You must save the configuration and reload the software to place the change into effect.

To enable MSDP, enter the following commands.

```
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# write memory
```

Syntax: [no] router msdp

Configuring MSDP peers

To configure an MSDP peer, enter a command such as the following at the MSDP configuration level.

```
BigIron RX(config-msdp-router)# 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.

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 subnet interface configured on the port. Also, make sure the IP address of the connect-source loopback is the same as the source IP address used by the MSDP router, the PIM-RP, and the BGP router.

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.

```
BigIron RX(config)# interface loopback 1
BigIron RX(config-lbif-1)# ip address 9.9.9.9/32
BigIron RX(config-lbif-1)# exit
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# msdp-peer 2.2.2.99 connect-source loopback 1
```

Designating an interface's IP address as the RP's IP address

When an RP receives a Source Active message, it checks its PIM Sparse multicast group table for receivers for the group. If it finds a receiver, the RP sends a Join message for that receiver back to the RP that originated the Source Active message. The originator RP is identified by its RP address.

By default, the IP address included in the RP address field of the SA message is the IP address of the originating RP, but 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's IP address to be the IP address of the RP, enter commands such as the following.

```
BigIron RX(config)# interface loopback 2
BigIron RX(config-lbif-2)# ip address 2.2.1.99/32
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# originator-id loopback 2
BigIron RX(config-msdp-router)# exit
```

Syntax: [no] originator-id <*type*> <*number*>

The **originator-id** parameter instructs MSDP to use the specified 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. There are no default originator-ids.

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. The filter only bars the DUT from local processing of the MSDP SAs.

- **sa-filter in** – Filters source-group pairs received in Source-Active messages from an MSDP neighbor.
- **sa-filter originate** – Filters source-group pairs in Source-Active messages in advertisements to an MSDP neighbor.

Filtering incoming 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 out (ignored) from local processing of the MSDP SAs.
- For peer 2.2.2.97, all source-group pairs except those with 10.x.x.x as the source are permitted.
- For peer 2.2.2.96, all source-group pairs except those associated with RP 2.2.42.3 are permitted.

Example

The following commands configure an IP address on port 3/1. This is the port on which the MSDP neighbors will be configured.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e1000-3/1)# ip address 2.2.2.98/24
BigIron RX(config-if-e1000-3/1)# exit
```

The following commands configure a loopback interface. The device will use this interface as the source address for communicating with the MSDP neighbors.

```
BigIron RX(config)# interface loopback 1
BigIron RX(config-lbif-1)# ip address 9.9.9.8/32
BigIron RX(config-lbif-1)# exit
```

The following commands configure extended ACLs. The ACLs will be used in route maps, which will be used by the Source-Active filters.

```
BigIron RX(config)# access-list 124 permit ip 10.0.0.0 0.255.255.255 any
BigIron RX(config)# access-list 124 permit ip host 2.2.2.2 any
BigIron RX(config)# access-list 125 permit ip any any
```

The following commands configure the route maps.

```
BigIron RX(config)# route-map msdp_map deny 1
BigIron RX(config-routemap msdp_map)# match ip address 123
BigIron RX(config-routemap msdp_map)# exit
BigIron RX(config)# route-map msdp2_map permit 1
BigIron RX(config-routemap msdp2_map)# match ip address 125
BigIron RX(config-routemap msdp2_map)# exit
BigIron RX(config)# route-map msdp2_rp_map deny 1
BigIron RX(config-routemap msdp2_rp_map)# match ip route-source 124
BigIron RX(config-routemap msdp2_rp_map)# exit
```

The following commands enable MSDP and configure the MSDP neighbors on port 3/1.

```
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# msdp-peer 2.2.2.99 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 2.2.2.97 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 2.2.2.96 connect-source loopback 1
BigIron RX(config-msdp-router)# exit
```

The following commands configure the Source-Active filters.

```
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# sa-filter in 2.2.2.99
BigIron RX(config-msdp-router)# sa-filter in 2.2.2.97 route-map msdp_map
BigIron RX(config-msdp-router)# 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 filters the DUT from local processing and 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 ignores source-group pairs received from neighbor 2.2.2.97 if the pairs have source address 10.x.x.x and any group address.
- **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 <*ip-addr*> [route-map <*map-tag*>] [rp-route-map <*rp-map-tag*>]

The <*ip-addr*> parameter specifies the IP address of the MSDP neighbor. The filter applies to Active-Source messages received from 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 and group 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 origin. If you use the **route-map** parameter instead, messages are filtered based on source-group pairs but not based on origin. Use the **match ip route-source** <*acl-id*> command in the route map to specify the RP address.

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 receive the matching source-group pairs. A deny action in the route map drops the matching source-group pairs.

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 an IP address on port 3/1. This is the port on which the MSDP neighbors will be configured.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e1000-e1000-3/1)# ip address 2.2.2.98/24
BigIron RX(config-if-e1000-3/1)# exit
```

The following commands configure a loopback interface. The device will use this interface as the source address for communicating with the MSDP neighbors.

```
BigIron RX(config)# interface loopback 1
BigIron RX(config-lbif-1)# ip address 9.9.9.8/32
BigIron RX(config-lbif-1)# exit
```

The following command configures an extended ACL to specify the source and group addresses you want to filter.

```
BigIron RX(config)# access-list 123 permit ip 10.0.0.0 0.255.255.255 any
```

The following commands configure a route map. The map matches on source address 10.x.x.x and any group address. Since the action is deny, the Source-Active filter that uses this route map will remove the source-group pairs that match this route map from the Source-Active messages to the neighbor.

```
BigIron RX(config)# route-map msdp_map deny 1
BigIron RX(config-routemap msdp_map)# match ip address 123
BigIron RX(config-routemap msdp_map)# exit
```

The following commands enable MSDP and configure MSDP neighbors on port 3/1.

```
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# msdp-peer 2.2.2.99 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 2.2.2.97 connect-source loopback 1
BigIron RX(config-if-3/1)# exit
```

The following commands configure the Source-Active filter.

```
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# sa-filter originate route-map msdp_map
```

This filter removes source-group pairs that match route map msdp_map from Source-Active messages before sending them to MSDP neighbors.

Syntax: [no] sa-filter originate [route-map <*map-tag*>]

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 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 receive the matching source-group pairs. A deny action in the route map drops the matching source-group pairs.

Displaying the differences before and after the source active filters are applied

This is an example of the Source Actives in the MSDP cache that will be displayed before the filter is applied.

```
BigIron RX #show ip msdp sa
Total 50 entries
Index  SourceAddr    GroupAddr               Age
1 (117.1.0.60, 224.200.1.40), RP:2.2.2.2, Age:0
2 (117.1.0.33, 224.200.1.13), RP:2.2.2.2, Age:0
3 (117.1.0.47, 224.200.1.27), RP:2.2.2.2, Age:0
4 (117.1.0.20, 224.200.1.0), RP:2.2.2.2, Age:0
5 (117.1.0.61, 224.200.1.41), RP:2.2.2.2, Age:0
6 (117.1.0.34, 224.200.1.14), RP:2.2.2.2, Age:0
7 (117.1.0.48, 224.200.1.28), RP:2.2.2.2, Age:0
8 (117.1.0.21, 224.200.1.1), RP:2.2.2.2, Age:0
9 (117.1.0.62, 224.200.1.42), RP:2.2.2.2, Age:0
10 (117.1.0.35, 224.200.1.15), RP:2.2.2.2, Age:0
11 (117.1.0.49, 224.200.1.29), RP:2.2.2.2, Age:0
12 (117.1.0.22, 224.200.1.2), RP:2.2.2.2, Age:0
13 (117.1.0.63, 224.200.1.43), RP:2.2.2.2, Age:0
14 (117.1.0.36, 224.200.1.16), RP:2.2.2.2, Age:0
15 (117.1.0.50, 224.200.1.30), RP:2.2.2.2, Age:0
16 (117.1.0.23, 224.200.1.3), RP:2.2.2.2, Age:0
17 (117.1.0.64, 224.200.1.44), RP:2.2.2.2, Age:0
18 (117.1.0.37, 224.200.1.17), RP:2.2.2.2, Age:0
19 (117.1.0.51, 224.200.1.31), RP:2.2.2.2, Age:0
20 (117.1.0.24, 224.200.1.4), RP:2.2.2.2, Age:0
```

```
21 (117.1.0.65, 224.200.1.45), RP:2.2.2.2, Age:0
22 (117.1.0.38, 224.200.1.18), RP:2.2.2.2, Age:0
23 (117.1.0.52, 224.200.1.32), RP:2.2.2.2, Age:0
24 (117.1.0.25, 224.200.1.5), RP:2.2.2.2, Age:0
25 (117.1.0.66, 224.200.1.46), RP:2.2.2.2, Age:0
26 (117.1.0.39, 224.200.1.19), RP:2.2.2.2, Age:0
27 (117.1.0.53, 224.200.1.33), RP:2.2.2.2, Age:0
28 (117.1.0.26, 224.200.1.6), RP:2.2.2.2, Age:0
29 (117.1.0.67, 224.200.1.47), RP:2.2.2.2, Age:0
30 (117.1.0.40, 224.200.1.20), RP:2.2.2.2, Age:0
31 (117.1.0.54, 224.200.1.34), RP:2.2.2.2, Age:0
32 (117.1.0.27, 224.200.1.7), RP:2.2.2.2, Age:0
33 (117.1.0.68, 224.200.1.48), RP:2.2.2.2, Age:0
34 (117.1.0.41, 224.200.1.21), RP:2.2.2.2, Age:0
35 (117.1.0.55, 224.200.1.35), RP:2.2.2.2, Age:0
36 (117.1.0.28, 224.200.1.8), RP:2.2.2.2, Age:0
37 (117.1.0.69, 224.200.1.49), RP:2.2.2.2, Age:0
38 (117.1.0.42, 224.200.1.22), RP:2.2.2.2, Age:0
39 (117.1.0.56, 224.200.1.36), RP:2.2.2.2, Age:0
40 (117.1.0.29, 224.200.1.9), RP:2.2.2.2, Age:0
41 (117.1.0.43, 224.200.1.23), RP:2.2.2.2, Age:0
42 (117.1.0.57, 224.200.1.37), RP:2.2.2.2, Age:0
43 (117.1.0.30, 224.200.1.10), RP:2.2.2.2, Age:0
44 (117.1.0.44, 224.200.1.24), RP:2.2.2.2, Age:0
45 (117.1.0.58, 224.200.1.38), RP:2.2.2.2, Age:0
46 (117.1.0.31, 224.200.1.11), RP:2.2.2.2, Age:0
47 (117.1.0.45, 224.200.1.25), RP:2.2.2.2, Age:0
48 (117.1.0.59, 224.200.1.39), RP:2.2.2.2, Age:0
49 (117.1.0.32, 224.200.1.12), RP:2.2.2.2, Age:0
50 (117.1.0.46, 224.200.1.26), RP:2.2.2.2, Age:0
Total number of SA Cache entries50
```

Syntax: show ip msdp sa

This is an example of the Source Actives in the MSDP cache that will be displayed after the filter is applied.

```
BigIron RX #show ip msdp sa
Total 6 entries
Index SourceAddr   GroupAddr              Age
1 (117.1.0.69, 224.200.1.49), RP:2.2.2.2, Age:0
2 (117.1.0.64, 224.200.1.44), RP:2.2.2.2, Age:0
3 (117.1.0.65, 224.200.1.45), RP:2.2.2.2, Age:0
4 (117.1.0.66, 224.200.1.46), RP:2.2.2.2, Age:0
5 (117.1.0.67, 224.200.1.47), RP:2.2.2.2, Age:0
6 (117.1.0.68, 224.200.1.48), RP:2.2.2.2, Age:0

Total number of SA Cache entries 6
```

Syntax: show ip msdp sa

Syntax: show ip msdp sa-cache

This display shows the following information.

TABLE 100 MSDP source active cache

This field...	Displays...
Total Entry	The total number of entries the cache can hold.
Used	The number of entries the cache currently contains.

TABLE 100 MSDP source active cache (Continued)

This field...	Displays...
Free	The number of additional entries for which the cache has room.
Index	The cache entry number.
SourceAddr	The IP address of the multicast source.
GroupAddr	The IP multicast group to which the source is sending information.
RP	The RP through which receivers can access the group traffic from the source
Age	The number of seconds the entry has been in the cache

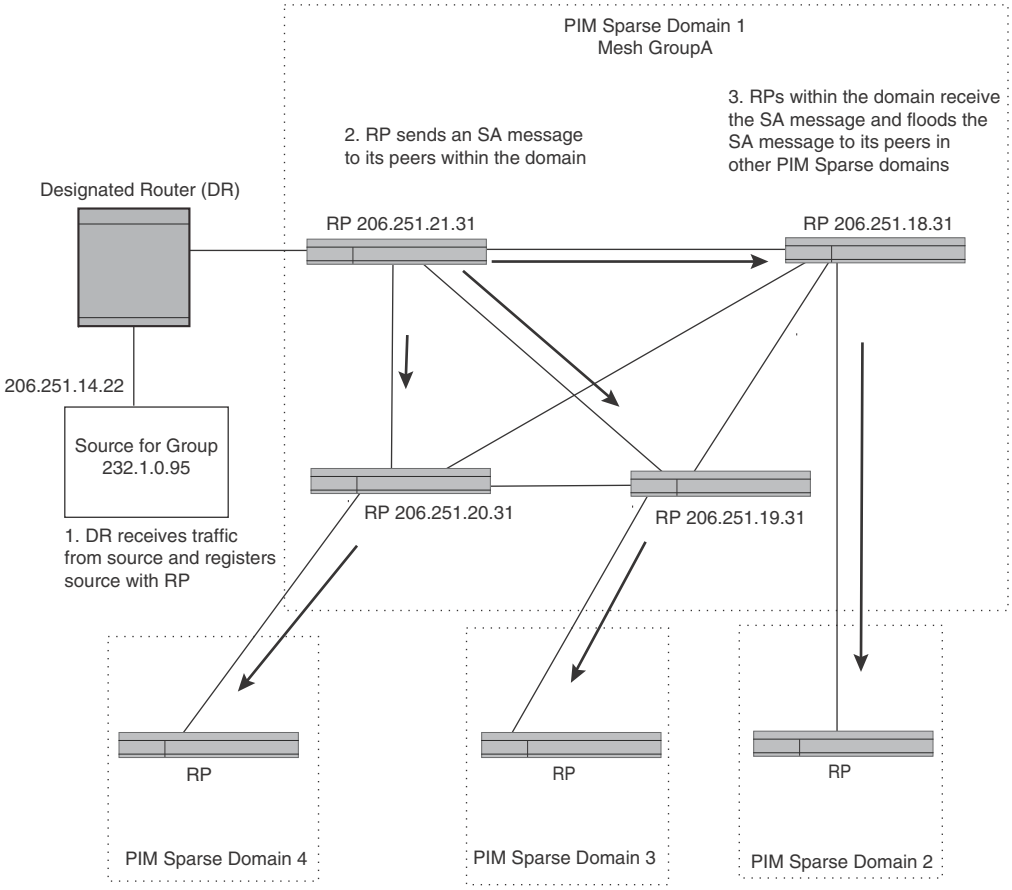
Configuring MSDP mesh groups

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 94.)

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 every 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 can forward the message to the members of a mesh group. If a mesh-group member receives a SA message from a MSDP peer that is not a member of the mesh-group, and the SA message passes the RPF check, then the member forwards the SA message to all members of the mesh-group. An RP can forward an SA message to any MSRP router as long as that peer is farther away from the originating RP than the current MSRP router.

Figure 94 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 94 Example of MSDP mesh group



PIM Sparse Domain 1 in Figure 94 contains a mesh group with four RPs. When the first RP, for example, RP 206.251.21.41 (which is also the originating RP), receives an SA message from the source, it sends the SA message to its peers within the domain, but the peers do not send the message back to the originator RP or to each other. The RPs then send the SA message to their peers in other domains.The process continues until all RPs within the network receive the SA message. RPs send join and prune messages to appropriate points on the multicast tree towards the originating RP.

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.

```
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# msdp-peer 163.5.34.10 connect-source loopback 2
BigIron RX(config-msdp-router)# msdp-peer 206.251.21.31 connect-source loopback 2
BigIron RX(config-msdp-router)# msdp-peer 206.251.17.31 connect-source loopback 2
BigIron RX(config-msdp-router)# msdp-peer 206.251.13.31 connect-source loopback 2
```

```
BigIron RX(config-msdp-router)# mesh-group GroupA 206.251.21.31
BigIron RX(config-msdp-router)# mesh-group GroupA 206.251.17.31
BigIron RX(config-msdp-router)# mesh-group GroupA 206.251.13.31
BigIron RX(config-msdp-router)# exit
```

Syntax: [no] mesh-group <*group-name*> <*peer-address*>

The sample configuration above reflects the configuration in Figure 94. On RP 206.251.21.31 you specify its peers within the same domain (206.251.21.31, 206.251.17.31, and 206.251.13.31).

You first configure the MSDP peers using the **msdp-peer** command to assign their IP addresses and the loopback interfaces. This information will be used as the source for sessions with the neighbor.

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 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 32 peers.

The **peer-address** parameter specifies the IP address of the MSDP peer that is being placed in the group.

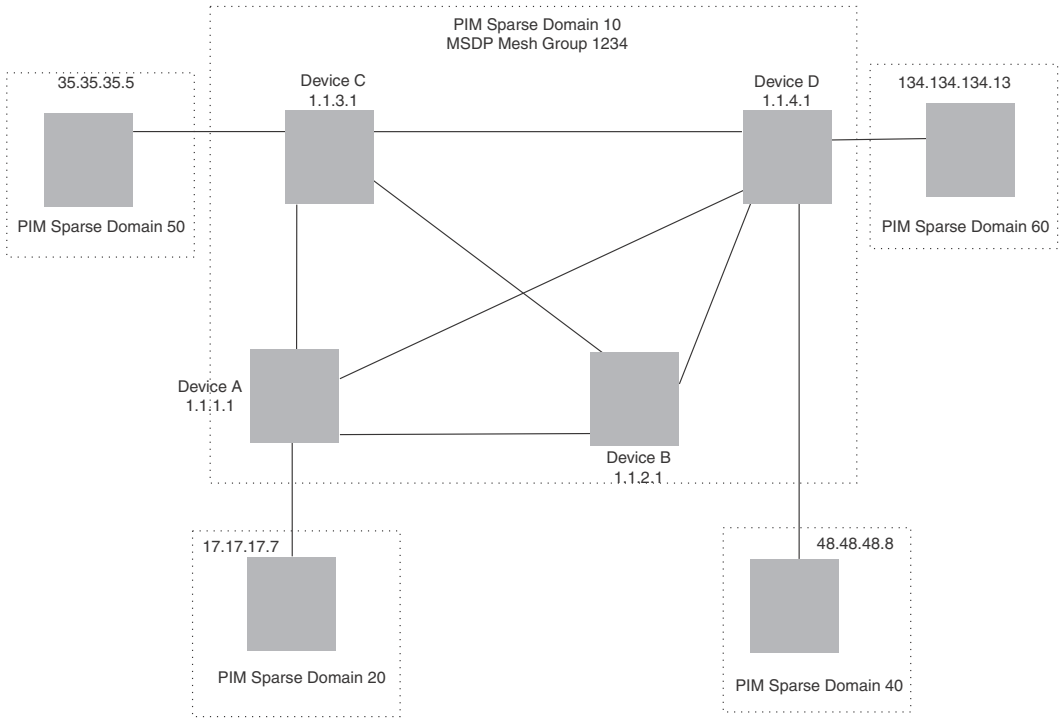
NOTE
On each of the device that will be part of the mesh-group, there must be a mesh-group definition for all the peers in the mesh-group.

Up to 32 MSDP peers can be configured per mesh group.

Example

In Figure 95, devices A, B, C, and D are in Mesh Group 1234. The example configuration following the figure shows how the devices are configured to be part of the MSDP mesh group. The example also shows the features that need to be enabled for the MSDP mesh group to work.

FIGURE 95 MSDP mesh group 1234



Configuration for Device A

The following set of commands configure the MSDP peers of Device A (1.1.1.1) that are inside and outside MSDP mesh group 1234. Device A's peers inside the mesh group 1234 are 1.1.2.1, 1.1.3.1, and 1.1.4.1. Device 17.17.17.7 is a peer of Device A, but is outside mesh group 1234. Multicast is enabled on Device A's interfaces. PIM and BGP are also enabled.

```
BigIron RX(config)# router pim
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 17.17.17.7
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.4.1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.3.1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.2.1
BigIron RX(config-msdp-router)# exit
BigIron RX(config)# interface loopback 1
BigIron RX(config-lbif-1)#ip address 1.1.1.1 255.255.255.0
BigIron RX(config-lbif-1)# ip pim-sparse
BigIron RX(config-lbif-1)# exit
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-1/1)# ip address 14.14.14.1 255.255.255.0
BigIron RX(config-if-1/1)# ip pim-sparse
BigIron RX(config-if-1/1)# exit
```

```
BigIron RX(config)# interface ethernet 2/1
BigIron RX(config-if-2/1)# ip address 12.12.12.1 255.255.255.0
BigIron RX(config-if-2/1)# ip pim-sparse
BigIron RX(config-if-2/1)# exit
BigIron RX(config)# interface ethernet 2/20
BigIron RX(config-if-2/20)# ip address 159.159.159.1 255.255.255.0
BigIron RX(config-if-2/20)# ip pim-sparse
BigIron RX(config-if-2/20)# exit
BigIron RX(config)# interface ethernet 4/1
BigIron RX(config-if-4/1)# ip address 31.31.31.1 255.255.255.0
BigIron RX(config-if-4/1)# ip pim-sparse
BigIron RX(config-if-4/1)# exit
BigIron RX(config)# interface ethernet 4/8
BigIron RX(config-if-4/8)# ip address 17.17.17.1 255.255.255.0
BigIron RX(config-if-4/8)# ip pim-sparse
BigIron RX(config-if-4/8)# ip pim border
BigIron RX(config-if-4/8)# exit
BigIron RX(config)# router pim
BigIron RX(config-router-pim)# bsr-candidate loopback 1 1 31
BigIron RX(config-router-pim)# rp-candidate loopback 1
BigIron RX(config-router-pim)# exit
BigIron RX(config)# router bgp
BigIron RX(config-bgp-router)# local-as 111
BigIron RX(config-bgp-router)# neighbor 31.31.31.3 remote-as 333
BigIron RX(config-bgp-router)# neighbor 31.31.31.3 next-hop-self
BigIron RX(config-bgp-router)# neighbor 12.12.12.2 remote-as 222
BigIron RX(config-bgp-router)# neighbor 12.12.12.2 next-hop-self
BigIron RX(config-bgp-router)# neighbor 14.14.14.4 remote-as 444
BigIron RX(config-bgp-router)# neighbor 14.14.14.4 next-hop-self
BigIron RX(config-bgp-router)# neighbor 17.17.17.7 remote-as 777
BigIron RX(config-bgp-router)# neighbor 17.17.17.7 next-hop-self
BigIron RX(config-bgp-router)# redistribute connected
BigIron RX(config-bgp-router)# write memory
```

Configuration for Device B

The following set of commands configure the MSDP peers of Device B. All Device B's peers (1.1.1.1, 1.1.3.1, and 1.1.4.1) are in the MSDP mesh group 1234. Multicast is enabled on Device B's interfaces. PIM and BGP are also enabled.

```
BigIron RX(config)# router pim
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.1.1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.3.1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.4.1
BigIron RX(config-msdp-router)# exit
BigIron RX(config)# interface loopback 1
BigIron RX(config-lbif-1)# ip address 1.1.2.1 255.255.255.0
BigIron RX(config-lbif-1)# ip pim-sparse
BigIron RX(config-lbif-1)# exit
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-1/1)# ip address 12.12.12.2 255.255.255.0
BigIron RX(config-if-1/1)# ip pim-sparse
BigIron RX(config-if-1/1)# exit
```

```
BigIron RX(config)# interface ethernet 1/12
BigIron RX(config-if-1/12)# ip address 165.165.165.1 255.255.255.0
BigIron RX(config-if-1/12)# ip pim-sparse
BigIron RX(config-if-1/12)# exit
BigIron RX(config)# interface ethernet 1/24
BigIron RX(config-if-1/24)# ip address 168.72.2.2 255.255.255.0
BigIron RX(config-if-1/24)# exit
BigIron RX(config)# interface ethernet 1/25
BigIron RX(config-if-1/25)# ip address 24.24.24.2 255.255.255.0
BigIron RX(config-if-1/25)# ip pim-sparse
BigIron RX(config-if-1/24)# exit
BigIron RX(config)# interface ethernet 8/1
BigIron RX(config-if-8/1)# ip address 32.32.32.2 255.255.255.0
BigIron RX(config-if-8/1)# ip pim-sparse
BigIron RX(config-if-1/24)# exit
BigIron RX(config)# router pim
BigIron RX(config-router-pim)# bsr-candidate loopback 1 2 32
BigIron RX(config-router-pim)# rp-candidate loopback 1
BigIron RX(config-router-pim)# exit
BigIron RX(config)# router bgp
BigIron RX(config-router-bgp)# local-as 222
BigIron RX(config-router-bgp)# neighbor 32.32.32.3 remote-as 333
BigIron RX(config-router-bgp)# neighbor 32.32.32.3 next-hop-self
BigIron RX(config-router-bgp)# neighbor 24.24.24.4 remote-as 444
BigIron RX(config-router-bgp)# neighbor 24.24.24.4 next-hop-self
BigIron RX(config-router-bgp)# neighbor 12.12.12.1 remote-as 111
BigIron RX(config-router-bgp)# neighbor 12.12.12.1 next-hop-self
BigIron RX(config-router-bgp)# redistribute connected
BigIron RX(config-router-bgp)# write memory
```

Configuration for Device C

The following set of commands configure the MSDP peers of Device C (1.1.3.1) that are inside and outside MSDP mesh group 1234. Device C's peers inside the mesh group 1234 are 1.1.1.1, 1.1.2.1, and 1.1.4.1. Device 35.35.35.5 is a peer of Device C, but is outside mesh group 1234. Multicast is enabled on Device C's interfaces. PIM and BGP are also enabled.

```
BigIron RX(config)# router pim
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# msdp-peer 35.35.35.5
BigIron RX(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.2.1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.1.1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.4.1
BigIron RX(config-msdp-router)# exit
BigIron RX(config)# interface loopback 1
BigIron RX(config-lbif-1)# ip address 1.1.3.1 255.255.255.0
BigIron RX(config-lbif-1)# ip pim-sparse
BigIron RX(config-lbif-1)# exit
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-3/1)# ip address 32.32.32.3 255.255.255.0
BigIron RX(config-if-3/1)# ip pim-sparse
BigIron RX(config-if-3/1)# exit
BigIron RX(config)# interface ethernet 10/1
BigIron RX(config-if-10/1)# ip address 31.31.31.3 255.255.255.0
BigIron RX(config-if-10/1)# ip pim-sparse
BigIron RX(config-if-10/1)# exit
```

```
BigIron RX(config)# interface ethernet 10/8
BigIron RX(config-if-10/8)# ip address 35.35.35.3 255.255.255.0
BigIron RX(config-if-10/8)# ip pim-sparse
BigIron RX(config-if-10/8)# ip pim border
BigIron RX(config-if-10/8)# exit
BigIron RX(config)# interface ethernet 12/2
BigIron RX(config-if-12/1)# ip address 34.34.34.3 255.255.255.0
BigIron RX(config-if-12/1)# ip pim-sparse
BigIron RX(config-if-12/1)# exit
BigIron RX(config)# interface ethernet 14/4
BigIron RX(config-if-14/4)# ip address 154.154.154.1 255.255.255.0
BigIron RX(config-if-12/1)# ip pim-sparse
BigIron RX(config-if-12/1)# exit
BigIron RX(config)# router pim
BigIron RX(config-router-pim)# bsr-candidate loopback 1 1 3
BigIron RX(config-router-pim)# rp-candidate loopback 1
BigIron RX(config-router-pim)# exit
BigIron RX(config)# router bgp
BigIron RX(config-router-bsr)# local-as 333
BigIron RX(config-router-bsr)# neighbor 35.35.35.5 remote-as 555
BigIron RX(config-router-bsr)# neighbor 35.35.35.5 next-hop-self
BigIron RX(config-router-bsr)# neighbor 32.32.32.2 remote-as 222
BigIron RX(config-router-bsr)# neighbor 32.32.32.2 next-hop-self
BigIron RX(config-router-bsr)# neighbor 34.34.34.4 remote-as 444
BigIron RX(config-router-bsr)# neighbor 34.34.34.4 next-hop-self
BigIron RX(config-router-bsr)# neighbor 31.31.31.1 remote-as 111
BigIron RX(config-router-bsr)# neighbor 31.31.31.1 next-hop-self
BigIron RX(config-router-bsr)# redistribute connected
BigIron RX(config-router-bsr)# write memory
```

Configuration for Device D

The following set of commands configure the MSDP peers of Device D (1.1.4.1) that are inside and outside MSDP mesh group 1234. Device D's peers inside the mesh group 1234 are 1.1.1.1, 1.1.2.1, and 1.1.3.1. Device 48.48.48.8 and 134.134.134.13 are also peers of Device D, but are outside mesh group 1234. Multicast is enabled on Device D's interfaces. PIM and BGP are also enabled.

```
BigIron RX(config)# router pim
BigIron RX(config)# router msdp
BigIron RX(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
BigIron RX(config-msdp-router)# msdp-peer 48.48.48.8
BigIron RX(config-msdp-router)# msdp-peer 134.134.134.13
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.1.1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.3.1
BigIron RX(config-msdp-router)# mesh-group 1234 1.1.2.1
BigIron RX(config-msdp-router)# exit
BigIron RX(config)# interface loopback 1
BigIron RX(config-lbif-)# ip address 1.1.4.1 255.255.255.0
BigIron RX(config-lbif-)# ip pim-sparse
BigIron RX(config-lbif-)# exit
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-)# ip address 24.24.24.4 255.255.255.0
BigIron RX(config-if-)# ip pim-sparse
BigIron RX(config-if-)# exit
```

```
BigIron RX(config)# interface ethernet 2/6
BigIron RX(config-if-)# ip address 156.156.156.1 255.255.255.0
BigIron RX(config-if-)# ip pim-sparse
BigIron RX(config-if-)# exit
BigIron RX(config)# interface ethernet 5/1
BigIron RX(config-if-)# ip address 34.34.34.4 255.255.255.0
BigIron RX(config-if-)# ip pim-sparse
BigIron RX(config-if-)# exit
BigIron RX(config)# interface ethernet 7/1
BigIron RX(config-if-)# ip address 14.14.14.4 255.255.255.0
BigIron RX(config-if-)# ip pim-sparse
BigIron RX(config-if-)# exit
BigIron RX(config)# interface ethernet 7/7
BigIron RX(config-if-)# ip address 48.48.48.4 255.255.255.0
BigIron RX(config-if-)# ip pim-sparse
BigIron RX(config-if-)# ip pim border
BigIron RX(config-if-)# exit
BigIron RX(config)# interface ethernet 7/8
BigIron RX(config-if-)# ip address 134.134.134.4 255.255.255.0
BigIron RX(config-if-)# ip pim-sparse
BigIron RX(config-if-)# ip pim border
BigIron RX(config-if-)# exit
BigIron RX(config)# router pim
BigIron RX(config-router-pim)# bsr-candidate loopback 1 14 34
BigIron RX(config-router-pim)# rp-candidate loopback 1
BigIron RX(config-router-pim)# exit
BigIron RX(config)# router bgp
BigIron RX(config-router-bsr)# local-as 444
BigIron RX(config-router-bsr)# neighbor 34.34.34.3 remote-as 333
BigIron RX(config-router-bsr)# neighbor 34.34.34.3 next-hop-self
BigIron RX(config-router-bsr)# neighbor 14.14.14.1 remote-as 111
BigIron RX(config-router-bsr)# neighbor 14.14.14.1 next-hop-self
BigIron RX(config-router-bsr)# neighbor 24.24.24.2 remote-as 222
BigIron RX(config-router-bsr)# neighbor 24.24.24.2 next-hop-self
BigIron RX(config-router-bsr)# neighbor 48.48.48.8 remote-as 888
BigIron RX(config-router-bsr)# neighbor 48.48.48.8 next-hop-self
BigIron RX(config-router-bsr)# neighbor 134.134.134.13 remote-as 1313
BigIron RX(config-router-bsr)# neighbor 134.134.134.13 next-hop-self
BigIron RX(config-router-bsr)# redistribute connected
BigIron RX(config-router-bsr)# write memory
```

Displaying MSDP information

You can display the following MSDP information:

- **Summary information** – the IP addresses of the peers, the state of the device's MSDP session with each peer, and statistics for Keepalive, Source Active, and Notification messages sent to and received from each of the peers.
- **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 device.

Displaying summary information

To display summary MSDP information, enter the following command at any level of the CLI.

```
BigIron RX# show ip msdp 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

This display shows the following information.

TABLE 101 MSDP summary information

This field...	Displays...
Peer Address	The IP address of the peer's interface with the device
State	The state of the MSDP router's 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 router has received from the peer
KA Out	The number of MSDP Keepalive messages the MSDP router has sent to the peer
SA In	The number of Source Active messages the MSDP router has received from the peer
SA Out	The number of Source Active messages the MSDP router 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, use the following CLI method.

```
BigIron RX# show ip msdp 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

This display shows the following information.

TABLE 102 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 router's 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 keep alive time, which specifies how often this MSDP router 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 router 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 Keep Alive messages the MSDP router has sent to the peer.

TABLE 102 MSDP peer information (Continued)

This field...	Displays...
Keep Alive Message Received	The number of Keep Alive messages the MSDP router has received from the peer.
Notifications Sent	The number of Notification messages the MSDP router has sent to the peer.
Notifications Received	The number of Notification messages the MSDP router has received from the peer.
Source-Active Sent	The number of Source Active messages the MSDP router has sent to the peer.
Source-Active Received	The number of Source Active messages the MSDP router has received from the peer.
Last Connection Reset Reason	The reason the previous session with this neighbor ended.
Notification Message Error Code Received	If the MSDP router receives a NOTIFICATION messages 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. • 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 error codes, see 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 102 MSDP peer 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.
Local host	The IP address of the MSDP router's 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 router 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 that the MSDP router 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.

TABLE 102 MSDP peer information (Continued)

This field...	Displays...
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 CLI method.

```
BigIron RX# show ip msdp sa-cache

Total Entry 4096, Used 1800 Free 2296
Index   SourceAddr  GroupAddr              Age
1     (100.100.1.254, 232.1.0.95), RP:206.251.17.41, Age:0
2     (100.100.1.254, 237.1.0.98), RP:206.251.17.41, Age:30
3     (100.100.1.254, 234.1.0.48), RP:206.251.17.41, Age:30
4     (100.100.1.254, 239.1.0.51), RP:206.251.17.41, Age:30
5     (100.100.1.254, 234.1.0.154), RP:206.251.17.41, Age:30
6     (100.100.1.254, 236.1.0.1), RP:206.251.17.41, Age:30
7     (100.100.1.254, 231.1.0.104), RP:206.251.17.41, Age:90
8     (100.100.1.254, 239.1.0.157), RP:206.251.17.41, Age:30
9     (100.100.1.254, 236.1.0.107), RP:206.251.17.41, Age:30
10    (100.100.1.254, 233.1.0.57), RP:206.251.17.41, Age:90
```

Syntax: show ip msdp sa-cache

This display shows the following information.

TABLE 103 MSDP source active cache

This field...	Displays...
Total Entry	The total number of entries the cache can hold.
Used	The number of entries the cache currently contains.
Free	The number of additional entries for which the cache has room.
Index	The cache entry number.
SourceAddr	The IP address of the multicast source.
GroupAddr	The IP multicast group to which the source is sending information.
RP	The RP through which receivers can access the group traffic from the source
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.

```
BigIron RX# clear ip msdp peer 205.216.162.1
Remote connection closed
```

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.

Clearing the source active cache

To clear the entries from the Source Active cache, enter the following command at the Privileged EXEC level of the CLI.

```
BigIron RX# clear ip msdp sa-cache
```

Syntax: clear ip msdp sa-cache [<*source-addr*> | <*group-addr*>]

The command in this example clears all the cache entries. Use the <*source-addr*> parameter to clear only the entries for a specified course. Use the <*group-addr*> parameter to clear only the entries for a specific group.

Clearing MSDP statistics

To clear MSDP statistics, enter the following command at the Privileged EXEC level of the CLI.

```
BigIron RX# 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 peer's IP address.

DVMRP overview

The device 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 96. 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 96, 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 97, 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 96 Downstream broadcast of IP multicast packets from source host

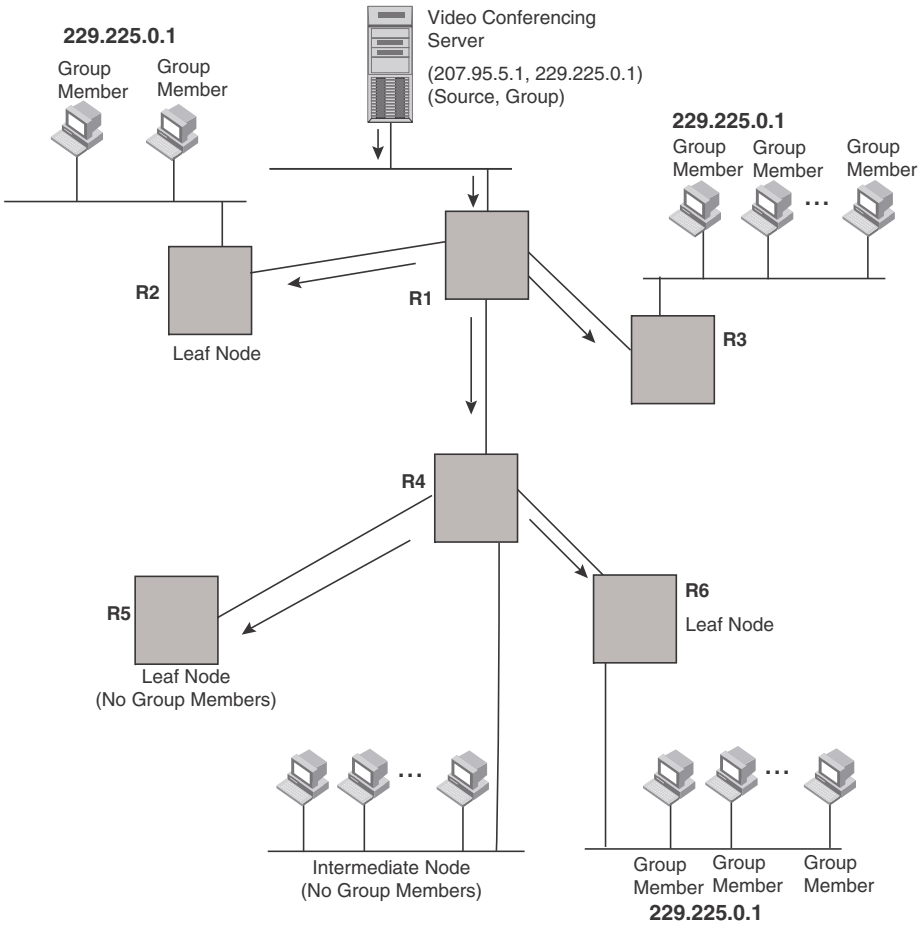
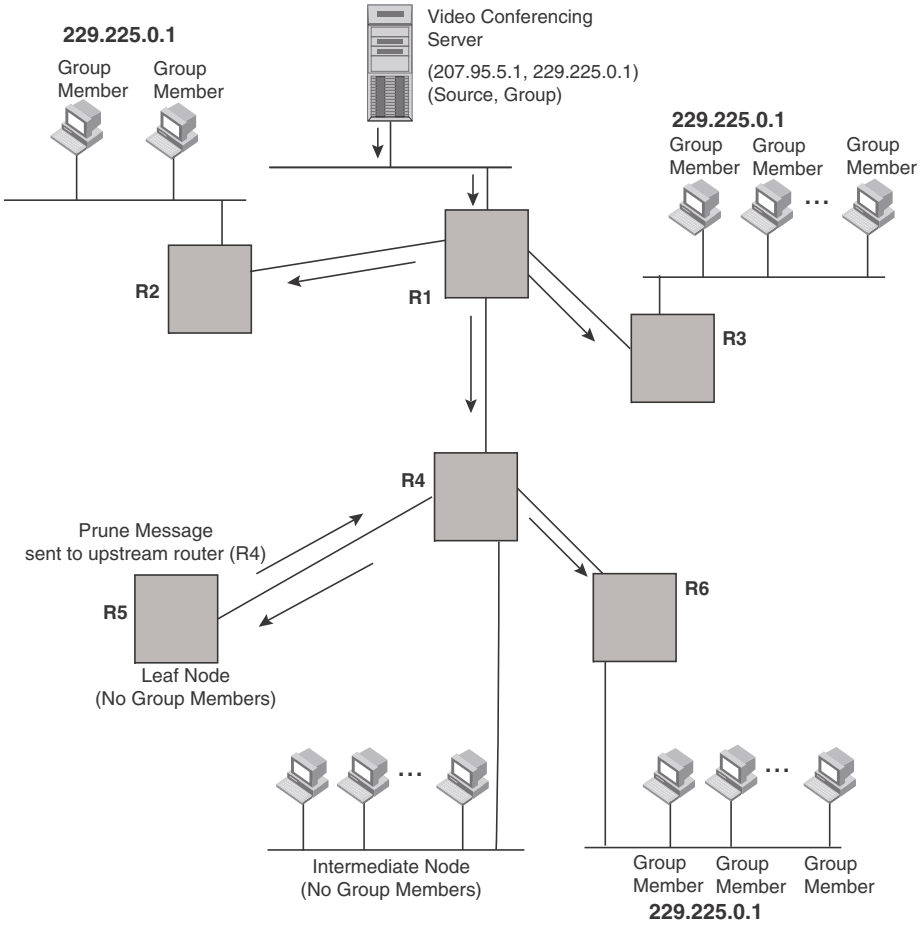


FIGURE 97 Pruning leaf nodes from a multicast tree



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 devicees that connect the various buildings need to be configured to support DVMRP multicasts from the designated video conference server as seen in Figure 96.

DVMRP is enabled on each of the device devices, shown in Figure 96, on which multicasts are expected. You can enable DVMRP on each device independently or remotely from one device by a Telnet connection. Follow the same steps for each router.

Globally enabling and disabling DVMRP

To globally enable DVMRP, enter the following command.

```
BigIron RX(config)# router dvmrp
BigIron RX(config)#
```

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 configuration for PIM multicast on a device (router pim level) only.

Globally enabling or disabling DVMRP without deleting multicast configuration

As stated above enter **no router dvmrp** removed PIM configuration. If you want to disable or enable DVMRP without removing PIM configuration, enter the following command.

```
BigIron RX(config)# router dvmrp
BigIron RX(config-pim-router)# disable-dvmrp
```

Syntax: [no] disable-dvmrp

Use the [no] version of the command to re-enable DVMRP.

Enabling DVMRP on an interface

After globally enabling DVMRP on a device, enable it on each interface that will support the protocol.

To enable DVMRP on Router 1 and interface 3, enter the following.

```
Router1(config)# router dvmrp
Router1(config-dvmrp-router)# int e 3/1
Router1(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 40 – 8000 seconds. The default value is 180 seconds.

To modify the neighbor timeout value to 100, enter the following.

```
BigIron RX(config-dvmrp-router)# nbr 100
```

Syntax: 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.

```
BigIron RX(config-dvmrp-router)# route-expire-timeout 50
```

Syntax: route-expire-timeout <*20-4000*>

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.

```
BigIron RX(config-dvmrp-router)# route-discard-timeout 150
```

Syntax: 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.

```
BigIron RX(config-dvmrp-router)# prune 25
```

Syntax: 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.

```
BigIron RX(config-dvmrp-router)# graft 120
```

Syntax: 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.

```
BigIron RX(config-dvmrp-router)# probe 10
```

Syntax: 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.

```
BigIron RX(config-dvmrp-router)# report 90
```

Syntax: 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.

```
BigIron RX(config-dvmrp-router)# trigger-interval 20
```

Syntax: 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.

```
BigIron RX(config-dvmrp-router)# default-gateway 192.35.4.1
```

Syntax: 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
- Metric
- Advertising

Modifying the TTL

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.

```
BigIron RX(config)# int e 1/4
BigIron RX(config-if-e10000-1/4)# ip dvmrp ttl 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.

```
BigIron RX(config)# interface e 3/5
BigIron RX(config-if-e10000-3/5)# ip dvmrp metric 15
```

Syntax: [no] ip dvmrp metric <*1-31*>

Enabling advertising

You can turn the advertisement of a local route on (enable) or off (disable) on the interface. By default, advertising is enabled.

To enable advertising on an interface, enter the following.

```
BigIron RX(config-if-e10000-1/4)# ip dvmrp advertise-local on
```

Syntax: [no] ip dvmrp advertise-local on | off

Displaying information about an upstream neighbor device

You can view information about the upstream neighbor device for a given source IP address for IP PIM packets. The software uses the IP route table or multicast route table to lookup the upstream neighbor device.

The following shows example messages that the Brocade device can display with this command.

```
BigIron RX# show ip dvmrp rpf 1.1.20.2|
directly connected or through an L2 neighbor
BigIron RX# show ip dvmrp rpf 1.2.3.4
no route
BigIron RX# show ip dvmrp rpf 1.10.10.24
upstream neighbor=1.1.20.1 on v21 using ip route
```

Syntax: show ip dvmrp rpf <*IP address*>

Where <*IP address*> is a valid source IP address

NOTE
If there are multiple equal cost paths to the source, the **show ip dvmrp rpf** command output may not be accurate. If your system has multiple equal cost paths, use the command **show ip dvmrp mcache** to view information about the upstream neighbor.

Configuring a static multicast route

The **ip mroute** command is used to direct multicast traffic along a specific path. The ip mroute command starts with the ip address or ingress ip address the source traffic is received upon. The ingress interface network mask, and the next hop address leading back to the ingress source ip address.

To configure static IP multicast routes, enter a command such as the following.

```
BigIron RX(config)# ip mroute 12.7.1.0 255.255.255.0 17.3.1.2
```

If you configure more than one static multicast route, the device Series router 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.

Syntax: [no] ip mroute <*ip-addr*> <*ip-mask*> [<*next-hop-ip-addr*> | ethernet <*slot/port*> | ve <*num*> | null0] [<*cost*>] [distance < *num*>]

The **ip-addr** and **ip-mask** parameters specifies the PIM source for the route.

The **ethernet** <*slot/port*> parameter specifies a physical port.

The **ve** <*num*> parameter specifies a virtual interface.

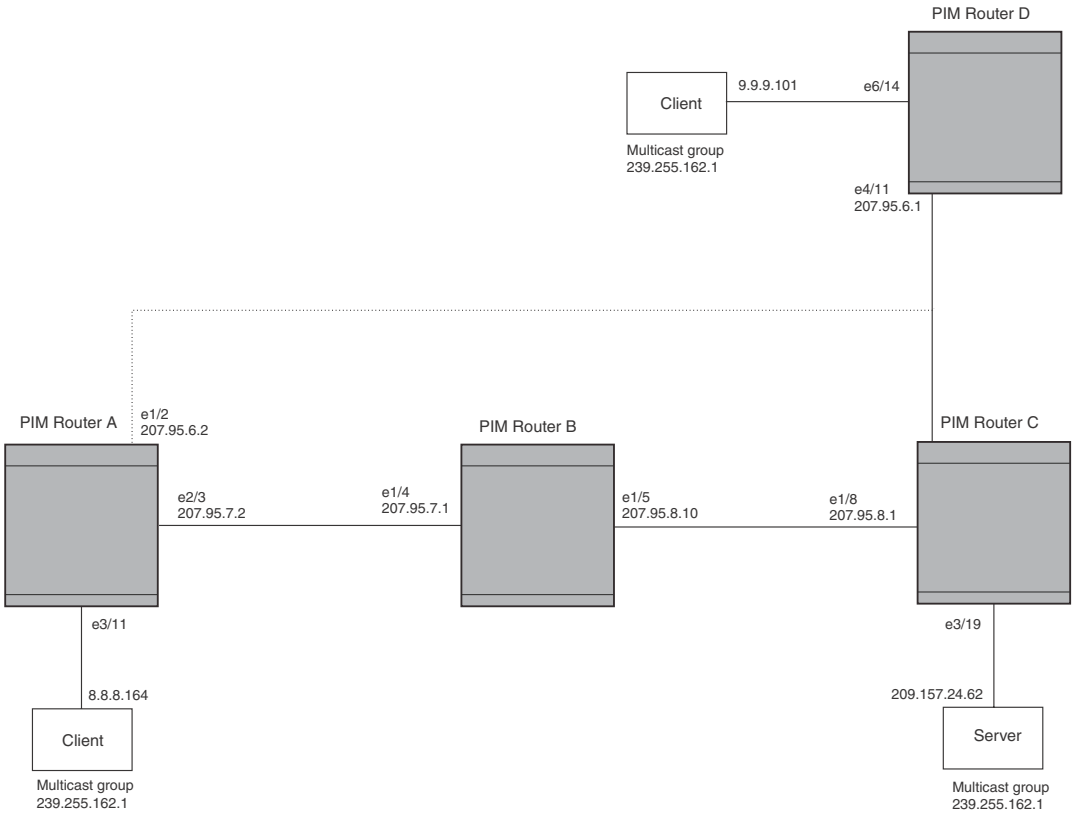
The **null0** parameter is the same as dropping the traffic.

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 device Series router always prefers directly connected routes over other routes.

FIGURE 98 Example multicast static routes



To add a static route to a virtual interface, enter commands such as the following.

```
BigIron RX(config)# ip mroute 3 0.0.0.0 0.0.0.0 ve 1 distance 1
BigIron RX(config)# write memory
```

Configuring IP multicast traffic reduction

In Layer 2 mode, device forwards all IP multicast traffic by default based on the Layer 2 information in the packets. Optionally, you can enable the device to make forwarding decisions in hardware, based on multicast group by enabling the IP Multicast Traffic Reduction feature.

NOTE
The IP Multicast Traffic Reduction feature is applicable for Layer 2 mode only.

When this feature is enabled, the device examines the MAC address in an IP multicast packet and forward the packet only on the ports from which the device has received Group Membership reports for that group, instead of forwarding all multicast traffic to all ports. The device sends traffic for other groups out all ports.

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 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.

  NOTE
  A router-id is required if a virtual interface (ve) or IP is not configured for IGMP snooping to work.

- **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 device.

Enabling IP multicast traffic reduction

By default, the device forwards all IP multicast traffic out all ports except the port on which the traffic was received. To reduce multicast traffic through the device, you can enable IP 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 IGMP 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 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, the device 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.

NOTE
When one or more device 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, configure all the device devices for passive IGMP and allow the to actively send the IGMP queries.

NOTE
A router-id is required if a virtual interface (ve) or IP is not configured for IGMP snooping to work.

To enable IP Multicast Traffic Reduction, enter the following command.

```
BigIron RX(config)# ip multicast active
```

Syntax: [no] ip multicast active | passive

When you enable IP multicast on a device, all ports on the device are configured for IGMP.

If you are using active IGMP, all ports can send IGMP queries and receive IGMP reports. If you are using passive IGMP, all ports can receive IGMP queries.

IP Multicast Traffic Reduction cannot be disabled on individual ports of a device. IP Multicast Traffic Reduction must can be disabled globally by entering the **no ip multicast** command.

NOTE
If the "route-only" feature is enabled on the device, then IP Multicast Traffic Reduction will not be supported.

To verify that IP Multicast Traffic Reduction is enabled, enter the following command at any level of the CLI.

```
BigIron RX(config)# show ip multicast
IP multicast is enabled - Active
```

Syntax: show ip multicast

Configuring the IGMP mode per VLAN

NOTE
A router-id is required if a virtual interface (ve) or IP is not configured for IGMP snooping to work.

If the IP Multicast command is not applied globally as described in "Enabling IP multicast traffic reduction" on page 653, you can apply it to individual VLANs instances within their configurations. In the following example, multicast traffic reduction is applied using IGMP snooping to VLAN 2.

```
(config)# vlan 2
(config-vlan-2)# multicast passive
```

Syntax: [no] multicast active | passive

When you enable IP multicast for a specific VLAN 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 instance. You can use active or passive IGMP mode. There is no default mode.

- **Active** – When active IGMP mode is enabled, the switch actively sends out IGMP queries to identify IP multicast groups within the VLAN 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 switch listens for IGMP Group Membership reports on the VLAN instance specified but does not send IGMP queries. The passive mode is called “IGMP snooping”. Use this mode when another device in the VLAN instance is actively sending queries.

Configuring IGMP snooping tracking per VLAN instance

When IGMP Snooping Tracking is enabled, the device immediately removes any GMP 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.

```
BigIron RX(config)# ip 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.

```
BigIron RX(config)# vlan 100
BigIron RX(config-vlan-100)# ip multicast tracking
```

Syntax: [no] ip multicast tracking

The **no** form of this command disables the tracking process per VLAN.

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] ip multicast tracking

The **no** form of this command disables the tracking process per VLAN instance.

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.

- **Active** – When active IGMP mode is enabled, a Brocade 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 network with no external IP multicast 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.

```
BigIron RX(config)# ip multicast active
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# reload
```

Syntax: [no] ip multicast active | passive

To enable passive IGMP, enter the following command.

```
BigIron RX(config)# ip multicast passive
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# reload
```

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 device 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.

```
BigIron RX(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 60 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.

```
BigIron RX(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 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 IP multicast filtering, enter the following command.

```
BigIron RX(config)# ip multicast filter
```

Syntax: [no] ip multicast filter

NOTE
If the "route-only" feature is enabled on a device, PIM SM traffic snooping will not be supported.

Layer 2 multicast filters

Beginning with release 02.6.00, you can define multicast boundaries on a per VLAN basis. The **multicast-boundary** command allows you to configure a boundary on IGMP snooping enabled interface by defining which multicast groups may not forward packets over a specified interface.

Configuration considerations

- Only one ACL can be bound to any interface or VLAN.
- If a new multicast boundary has to be applied, you must delete the old bounder first, then apply the new ACL.
- To avoid temporary loss in multicast traffic, ACLs should be configured before applying them to multicast boundaries.
- Modifying an already applied ACL will take effect immediately.
- Configurations shoube be generated at the VLAN level if user has explicitly configured it, regardless of whether it matches the global snooping configuration.
- You can issue the "no multicast" command to erase all VLAN-level configuration and force the VLAN to inherit the global configuration.
- You can issue the **[no] multicast active|passive** command to explicitly set the VLAN-level configuration.

  NOTE
  Issuing the **[no] multicast active|passive** command causes this configuration to always be generated for this VLAN.

- Global configurations will not affect any explicit VLAN-level configurations.

Configuring Layer 2 multicast boundaries

You can define multicast boundaries on a per VLAN basis by entering commands such as the following.

```
BigIron RX(config)#vlan 100
BigIron RX(config-vlan-100)#multicast-boundary MyFoundryAccessList ethernet 3/22
```

Syntax: [no] multicast-boundary <*acl-spec*> [ethernet <*slot/port*>]

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 **port-list** parameter to define the member ports on which the ACL is applied. The ACL will be applied to the multicast traffic arriving in both directions.

Use the **no multicast boundary** command to remove the boundary on an IGMP enabled interface.

NOTE
The ACL, MyFoundryAccessList can be configured using standard ACL syntax which can be found in the ACL section.

PIM SM traffic snooping

By default, when a device 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 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 IP multicast group traffic only on the ports that are attached to receivers for the group.

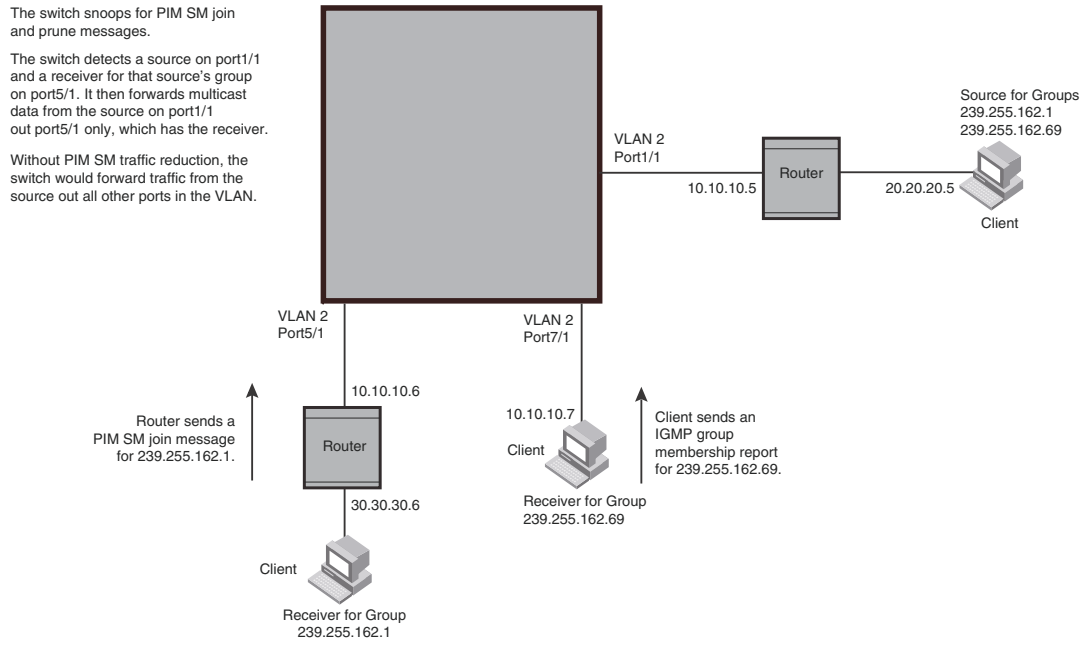
PIM SM traffic snooping requires IP multicast traffic reduction to be enabled on the device. IP multicast traffic reduction configures the device to listen for IGMP messages. PIM SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for 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 99 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 99 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.

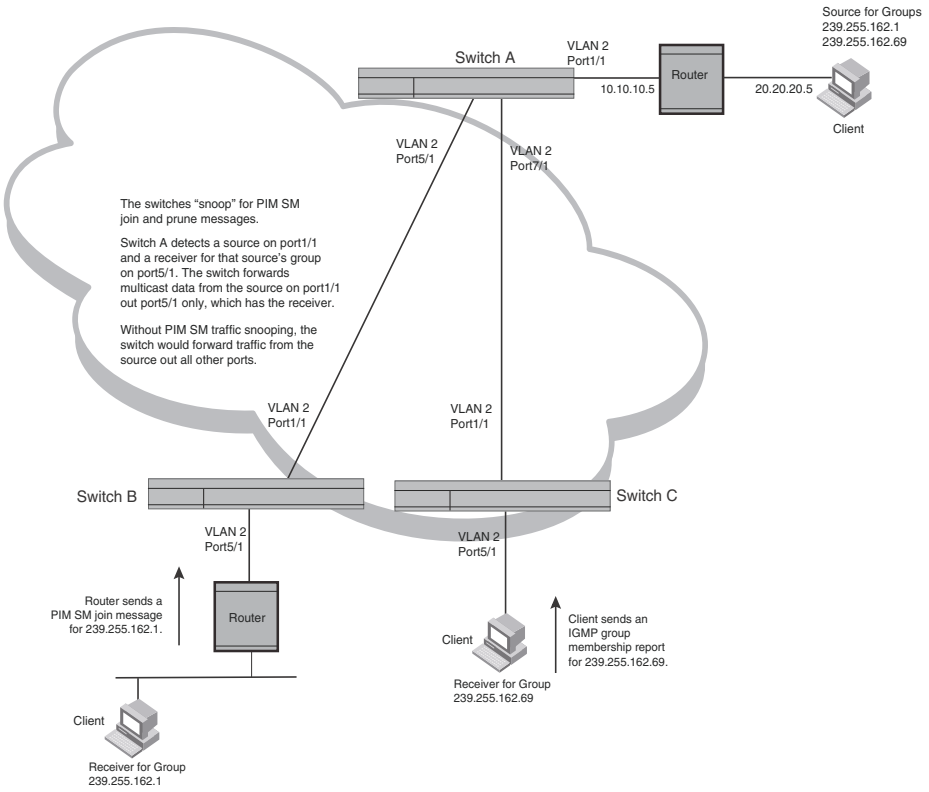
The IP multicast traffic reduction feature and the PIM SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM SM groups learned through join messages as well as MAC addresses learned through IGMP group membership reports. In this case, even 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 99.

Figure 100 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 NetIron's.

FIGURE 100 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

- 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 device, 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.

```
BigIron RX(config)# ip multicast
BigIron RX(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.

```
BigIron RX(config)# no ip pimsm-snooping
```

If you also want to disable IP multicast traffic reduction, enter the following command.

```
BigIron RX(config)# no ip multicast
```

Configuring the PIM SM traffic snooping per VLAN instance

If PIM SM Traffic snooping is not applied globally, you can apply it to individual VLANs instances within their configurations. In the following example, multicast traffic reduction is applied using PIM SM Traffic snooping to VLAN 2.

```
(config)# vlan 2
(config-vlan-2)# multicast pimsm-snooping
```

Syntax: [no] multicast pimsm-snooping

Configuring PIM proxy per VLAN instance

Using the PIM proxy function, multicast traffic can be reduced by configuring an BigIron RX switch to issue PIM join and prune messages on behalf of hosts that the configured switch discovers through standard PIM interfaces. The switch 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 switches, this removes the need to send multiple messages.

To configure an BigIron RX switch to function as a PIM proxy on VLAN 2, use the following commands.

```
(config)# vlan 2
(config-vlan-2)# multicast pim-proxy-enable
```

Syntax: [no] multicast pim-proxy-enable

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 switch, 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 switch. The snooping device will respond to IGMP queries from the uplink multicast PIM switch for the groups and sources configured. Upon the multicast switch 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 switch ports or dropped in hardware if no other multicast receiver and switches are present in the VLAN.

The **multicast static-group uplink** command can be configured under the VLAN configuration only.

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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX((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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX(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 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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX(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 switch 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.

Chapter

24

Configuring RIP

In this chapter

Overview of Routing Information Protocol (RIP)

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 BigIron RX 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 BigIron RX.

RIP routers, including the device, 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.

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.

```
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(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 BigIron RX 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 device port increases the cost of a RIP route that is learned on the port. The device 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.

```
BigIron RX(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 765 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.

```
BigIron RX(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 redisctribution filters, enter a command such as the following.

```
BigIron RX(config-rip-router)#redistribute bgp route-map longroute
```

Syntax: 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.

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.

```
BigIron RX(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 BigIron RX 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.

```
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e10000-1/1)# ip rip learn-default
```

Syntax: [no] ip rip learn-default

Configuring a RIP neighbor filter

By default, a BigIron RX 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.

```
BigIron RX(config-rip-router)# neighbor 1 deny any
```

Syntax: [no] neighbor <*filter-num*> permit | deny <*source-ip-address*> | any

This command configures the device 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.

```
BigIron RX(config-rip-router)# neighbor 2 deny 192.16.1.170
BigIron RX(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.
- **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. This is the default.

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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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 BigIron RX for Virtual Router Redundancy Protocol (VRRP) or VRRP Extended (VRRPE). Refer to Chapter 17, "Configuring VRRP and VRRPE".

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.

```
BigIron RX(config)# ip prefix-list list1  permit 192.53.4.1 255.255.255.0
BigIron RX(config)# ip prefix-list list2  permit 192.53.5.1 255.255.255.0
BigIron RX(config)# ip prefix-list list3  permit 192.53.6.1 255.255.255.0
BigIron RX(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: 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.

```
BigIron RX(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.

```
BigIron RX(config-if-e1000-1/2)# ip rip prefix-list list2 in
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config) router rip
BigIron RX(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 filters

To display RIP filters, enter the following command at any CLI level.

```
BigIron RX> 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 30, Next Update 29
       Need trigger update 0, next trigger broadcast 1
       Minimum update interval 25, Max update offset 5
       Split horizon is on; poison reverse is off
       import metric 1
       Default routes are accepted
       Prefix List, Inbound, Not set
       Prefix List, Outbound, Not set
       Redistribute: CONNECTED  Metric : 0   Routemap : Not Set
Static Metric : 1  Routemap : map1   .not defined.
OSPF Metric : 1  Routemap  : Not Set

   RIP Neighbor Filter Table
   Index   Action   Neighbor IP Address
   1       permit   any
```

Syntax: show ip rip

This display shows the following information.

TABLE 104 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.

TABLE 104 CLI display of neighbor filter information (Continued)

This field...	Displays...
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.
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.

Clearing the RIP routes from the routing table

Clearing all the routes from the routing table.

To clear RIP local routes, enter a command such as the following.

```
BigIron(config)#clear ip rip local routes
```

Syntax: clear ip rip local routes

To clear the RIP routes from the RIP database, enter a command such as the following.

```
BigIron(config)# clear ip rip routes
```

Syntax: clear ip rip routes <*ip-addr*> / <*mask-bits*>

Use the **ip address** to specify which routes in the database you want to clear.

Use the **subnet mask** to specify which subnets you want to clear.

NOTE
Using the **clear ip route** command will not clear routes learned through RIP.

Chapter

Configuring OSPF Version 2 (IPv4) 25

In this chapter

Overview of OSPF (Open Shortest Path First)

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 device 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.

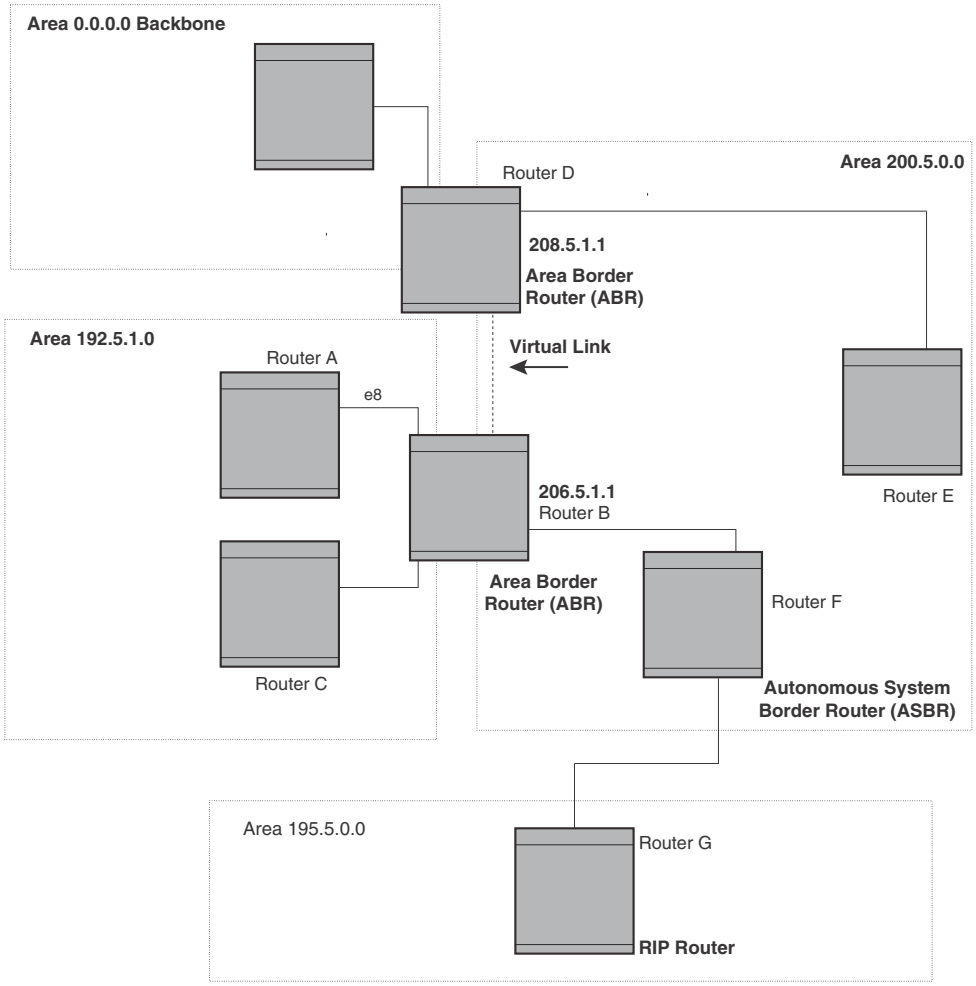
An AS can be divided into multiple ***areas*** as shown in Figure 101 on page 676. Each area represents a collection of contiguous networks and hosts. Areas limit the area to which link-state advertisements are broadcast, thereby limiting the amount of flooding that occurs within the network. An area 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 702.

FIGURE 101 OSPF operating in a network



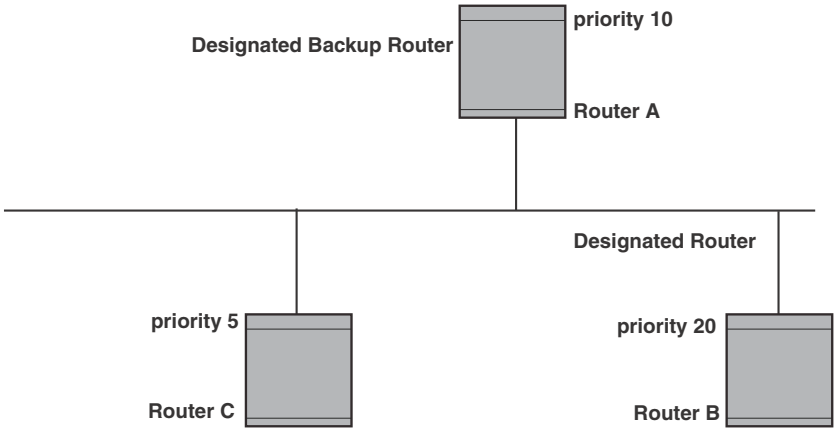
Designated routers in multi-access networks

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 102

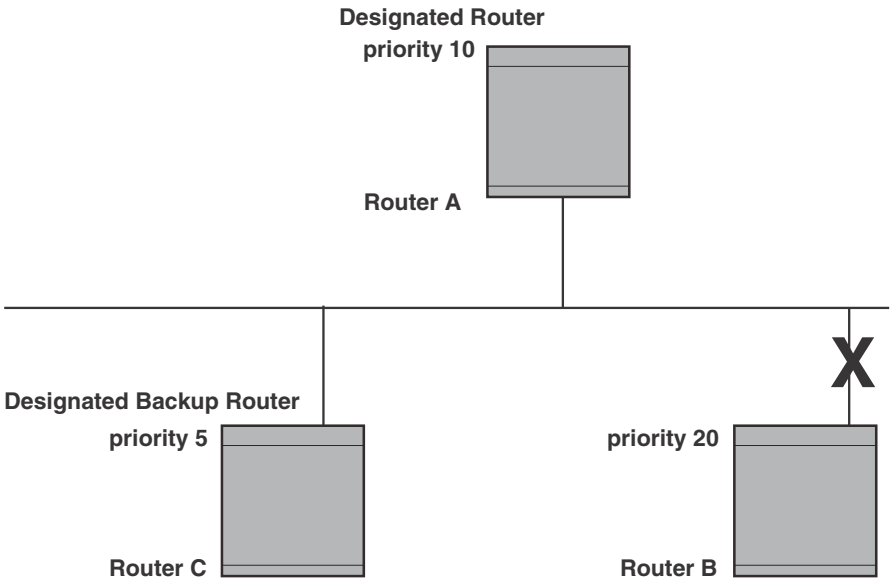
FIGURE 102 Designated and backup router election



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 103.

NOTE
Priority is a configurable option at the interface level. You can use this parameter to help bias one router as the DR.

FIGURE 103 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 Brocade router ID is the IP address configured on the lowest numbered loopback interface. If the BigIron RX does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, refer to "Changing the router ID" on page 174.

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

Brocade routers are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification. Brocade 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 716.

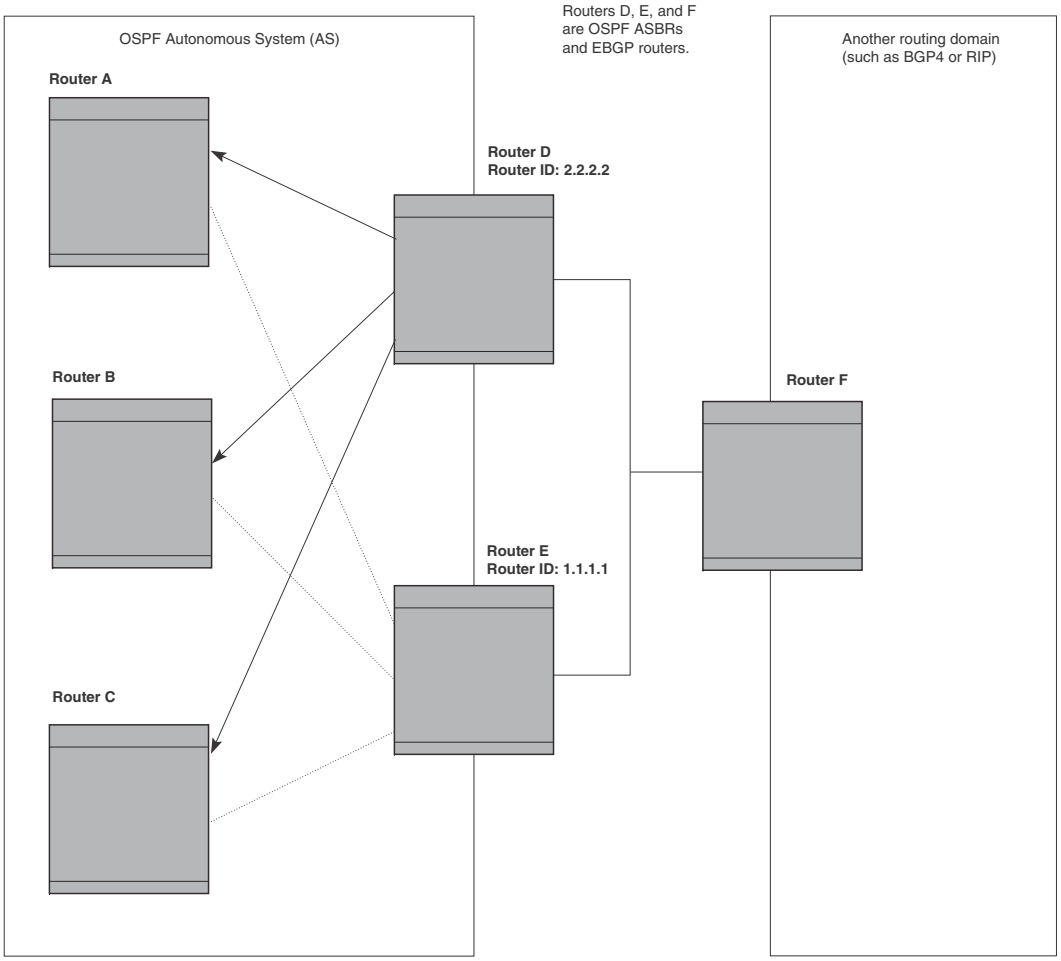
Reduction of equivalent AS external LSAs

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 BigIron RX's link state database. The AS External LSA reduction is described in RFC 2328

Figure 104 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 104 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 BigIron RX switches 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 device switches that flush the duplicate AS External LSAs have more memory for other OSPF data. In Figure 104, 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 104 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

BigIron RX provides support for 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 follows.

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.

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

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

- 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

- 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.

Interface parameters

- 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.

```
BigIron RX(config)# router ospf
```

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.

```
BigIron RX(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 101 on page 676, use the following method.

```
BigIron RX(config-ospf-router)# area 192.5.1.0
BigIron RX(config-ospf-router)# area 200.5.0.0
BigIron RX(config-ospf-router)# area 195.5.0.0
BigIron RX(config-ospf-router)# area 0.0.0.0
BigIron RX(config-ospf-router) write memory
```

Syntax: [no] area <*num*> | <*ip-addr*>

The <*num*> | <*ip-addr*> parameter specifies the area number, which can be a number or in IP address format. If you specify an number, the number can be from 0 – 2,147,483,647.

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 BigIron RX 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.

This feature does not apply to Not So Stubby Areas (NSSAs).

To disable summary LSAs for a stub area, enter commands such as the following.

```
BigIron RX(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)

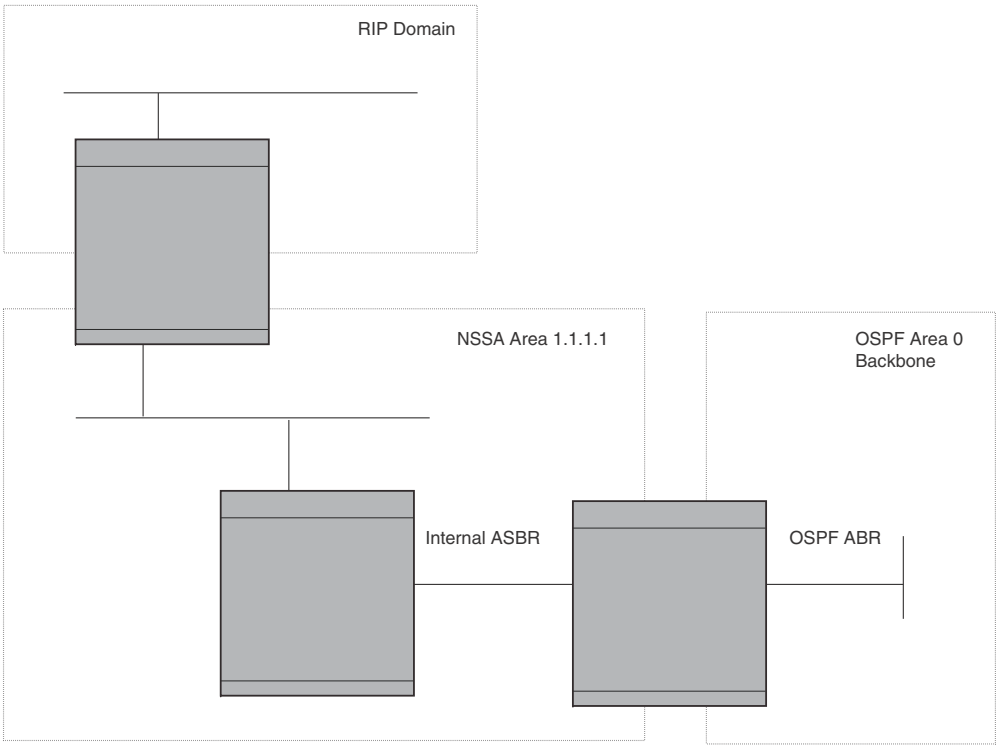
The OSPF Not So Stubby Area (NSSA) feature enables you to configure OSPF areas that provide the benefits of stub areas, but that also are capable of importing external route information. OSPF does not flood external routes from other areas into an NSSA, but does translate and flood route information 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 Brocade implementation of NSSA is based on RFC 3101.

Figure 105 shows an example of an OSPF network containing an NSSA.

FIGURE 105 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.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)# area 1.1.1.1 nssa 1
BigIron RX(config-ospf-router)# write memory
```

Syntax: area <*num*> | <*ip-addr*> nssa <*cost*> | 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 an 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.

NOTE
The BigIron RX 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.

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.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)# area 1.1.1.1 range 209.157.22.1 255.255.0.0
BigIron RX(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.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)# area 192.45.5.1 range 193.45.0.0 255.255.0.0
BigIron RX(config-ospf-router)# area 193.45.6.2 range 193.45.0.0 255.255.0.0
```

Syntax: 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-ospf-router)# 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 [0 | 1] <*string*>
- ip ospf cost <*num*>
- ip ospf dead-interval <*value*>
- ip ospf hello-interval <*value*>
- ip ospf md5-authentication key-activation-wait-time <*num*> | key-id <*num*> [0 | 1] key <*string*>
- 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	OSPF supports three methods of authentication for each interface—none, simple password, and MD5. Only one method of authentication can be active on an interface at a time. The default authentication value is none, meaning no authentication is performed. • The simple password method of authentication requires you to configure an alphanumeric password on an interface. The simple password setting takes effect immediately. All OSPF packets transmitted on the interface contain this password. Any OSPF packet received on the interface is checked for this password. If the password is not present, then the packet is dropped. The password can be up to eight characters long. • The MD5 method of authentication requires you to configure a key ID and an MD5 Key. The key ID is a number from 1 – 255 and identifies the MD5 key that is being used. The MD5 key can be up to sixteen alphanumeric characters long.
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.
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. By default, the dead timer interval is four times the hello timer interval. The default is 40 seconds.
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. On NBMA, the default is 30 seconds.
MD5-authentication activation wait time	The number of seconds the BigIron RX 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 ID and key	A method of authentication that requires you to configure a key ID and an MD5 key. The key ID is a number from 1 – 255 and identifies the MD5 key that is being used. The MD5 key consists of up to 16 alphanumeric characters. The MD5 is encrypted and included in each OSPF packet transmitted.

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. Refer to "Assigning an IP address to an Ethernet port" on page 154.
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 BigIron RX 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.

Encrypted display of the authentication string or MD5 authentication key

The Brocade implementation of OSPF authentication is based on RFC 2328. The optional 0 | 1 parameter with the **authentication-key** and **md5-authentication key-id** parameters affects encryption.

For added security, the device encrypts display of the password or authentication string. Encryption is enabled by default. The software also provides an optional parameter to disable encryption of a password or authentication string, on an individual OSPF area or OSPF interface basis.

When encryption of the passwords or authentication strings is enabled, they are encrypted in the CLI regardless of the access level you are using. The encryption option can be omitted (the default) or can be one of the following:

- **0** – Disables encryption for the password or authentication string you specify with the command. The password or string is shown as clear text in the running configuration and the startup configuration file. Use this option if you do not want display of the password or string to be encrypted.
- **1** – Assumes that the password or authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE
If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

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.

```
BigIron RX(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.

After you apply filters to block the outbound LSAs, the filtering occurs during the database synchronization and flooding.

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.

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.

```
BigIron RX(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 out

To remove the filter, enter a command such as the following.

```
BigIron RX(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.

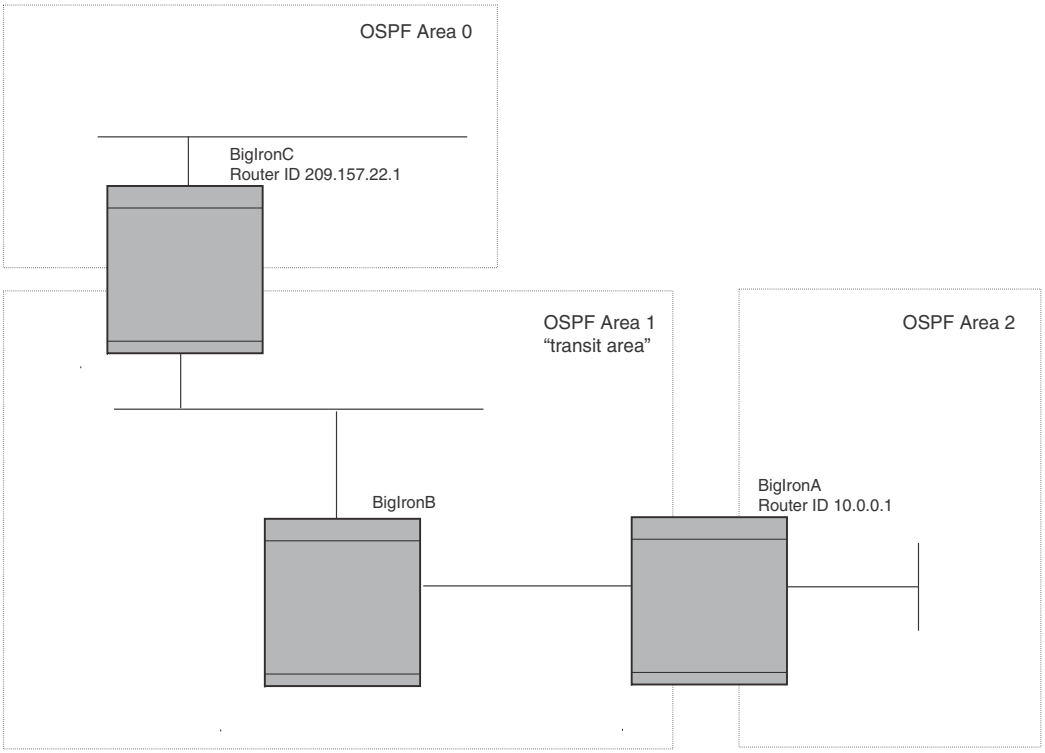
Two parameters fields must be defined for all virtual links—transit area ID and neighbor router:

- 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 Brocade router ID is the IP address configured on the lowest numbered loopback interface. If the BigIron RX does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, refer to "Changing the router ID" on page 174.

NOTE
When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

FIGURE 106 Defining OSPF virtual links within a network



Example

Figure 106 shows an OSPF area border router, BigIron RXA, that is cut off from the backbone area (area 0). To provide backbone access to BigIron RXA, you can add a virtual link between BigIron RXA and BigIron RXC 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 BigIron RXA, enter the following commands.

```
BigIron RXA(config)#router ospf
BigIron RXA(config-ospf-router)# area 2
BigIron RXA(config-ospf-router)# area 1
BigIron RXA(config-ospf-router)# area 1 virtual-link 209.157.22.1
BigIron RXA(config-ospf-router)# write memory
```

Enter the following commands to configure the virtual link on BigIron RXC.

```
BigIron RXC(config)#router ospf
BigIron RXC(config-ospf-router)# area 0
BigIron RXC(config-ospf-router)# area 1
BigIron RXC(config-ospf-router)# area 1 virtual-link 10.0.0.1
```

Syntax: [no] area <*ip-addr*> | <*num*> virtual-link <*router-id*>
[authentication-key | dead-interval | hello-interval | retransmit-interval | transmit-delay

<*value*> |
[md5-authentication key-activation-wait-time <*num*> | key-id <*num*> [0 | 1] key <*string*>]

The **area** <*ip-addr*> | <*num*> parameter specifies the transit area.

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 BigIron RX, enter the **show ip** command.

Refer to "Modify virtual link parameters" on page 694 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 <*num*> | <*ip-addr*> virtual-link <*ip-addr*> [authentication-key [0 | 1] <*string*>]
[dead-interval <*num*>]
[hello-interval <*num*>] [md5-authentication key-activation-wait-time <*num*> | key-id <*num*> [0 | 1] key <*string*>]
[retransmit-interval <*num*>] [transmit-delay <*num*>]

The parameters are described below.

Virtual link parameter descriptions

You can modify the following virtual link interface parameters.

Authentication Key	This parameter allows you to assign different authentication methods on a port-by-port basis. OSPF supports three methods of authentication for each interface—none, simple password, and MD5. Only one method of authentication can be active on an interface at a time. The simple password method of authentication requires you to configure an alphanumeric password on an interface. The password can be up to eight characters long. The simple password setting takes effect immediately. 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. The MD5 method of authentication encrypts the authentication key you define. The authentication is included in each OSPF packet transmitted.
MD5 Authentication Key	When simple authentication is enabled, the key is an alphanumeric password of up to eight characters. When MD5 is enabled, the key 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.
MD5 Authentication Key ID	The Key ID 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.
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.
Hello Interval	The length of time between the transmission of hello packets. The range is 1 – 65535 seconds. The default is 10 seconds. On NBMA, the default is 30 seconds.
Retransmit Interval	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	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.

Configuring an OSPF non-broadcast interface

OSPF routers generally use broadcast packets to establish neighbor relationships and broadcast route updates on Ethernet and virtual routing interfaces (ves). Beginning with BigIron RX software releases 02.3.00, you can configure an interface to send OSPF unicast packets rather than broadcast packets to its neighbor by configuring non-broadcast multi-access (NBMA) networks.

NBMA networks are similar to broadcast networks except the packets are sent as unicast. This type of network can be useful in situations where multicast traffic is not feasible (for example when a firewall does not allow multicast packets).

You configure NBMAs on an interface. The routers at the other end of that interface must have a non-broadcast neighbor configured. There is no restriction on the number of routers sharing a non-broadcast interface (for example, through a hub or switch).

To configure NBMA on an interface, do the following.

1. Create an OSPF area on an interface, then enable NBMA on that interface.

```
BigIron RX(config)# int ve 20
BigIron RX(config-vif-20)# ip ospf area 0
BigIron RX(config-vif-20)# ip ospf network non-broadcast
BigIron RX(config-vif-20)# exit
```

   Syntax: [no] ip ospf network non-broadcast

2. Then under the router OSPF level, specify the IP address of the neighbor in the OSPF configuration. The non-broadcast interface configuration must be done on the OSPF routers on both ends of the link.

   For example, the following commands configure VE 20 as a non-broadcast interface.

   The following commands specify 1.1.20.1 as an OSPF neighbor address. The address specified must be in the same sub-net as a non-broadcast interface.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)# neighbor 1.1.20.1
```

   Syntax: neighbor <*ip-address*>

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 both of the other routers must be specified as neighbors.

The output of the **show ip ospf interface** command has been enhanced to display information about non-broadcast interfaces and neighbors that are configured in the same sub-net.

For example.

```
BigIron RX# show ip ospf interface
v20,OSPF enabled
     IP Address 1.1.20.4, Area 0
     OSPF state BD, Pri 1, Cost 1, Options 2, Type nbma Events 6
     Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
     DR:  Router ID 1.1.13.1          Interface Address 1.1.20.5
     BDR: Router ID 2.2.2.1           Interface Address 1.1.20.4
     Neighbor Count = 1, Adjacent Neighbor Count= 2
     Non-broadcast neighbor config: 1.1.20.1, 1.1.20.2, 1.1.20.3, 1.1.20.5,
     Neighbor:         1.1.20.5
     Authentication-Key:None
     MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

In the Type field, “non-broadcast” indicates that this is a non-broadcast interface. When the interface type is non-broadcast, the Non-broadcast neighbor config field displays the neighbors that are configured in the same sub-net. If no neighbors are configured in the same sub-net, a message such as the following is displayed.

```
***Warning! no non-broadcast neighbor config in 1.1.100.1 255.255.255.0
```

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.

NOTE
This feature is supported in Ironware software for BigIron RX releases 02.2.01 and later.

NOTE
This feature is supported on Gigabit Ethernet and 10-Gigabit Ethernet interfaces.

NOTE
This feature is supported on physical interfaces. It is not supported on virtual interfaces.

NOTE
Brocade supports numbered point-to-point networks, meaning the OSPF router must have an IP interface address which uniquely identifies the router over the network. Brocade does not support unnumbered point-to-point networks.

Configuring an OSPF point-to-point link

To configure an OSPF point-to-point link, enter commands such as the following.

```
BigIron RX(config)# interface eth 1/5
BigIron RX(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

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.

```
BigIron RX# 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 105 Output of the show ip ospf interface command

This field	Displays
IP Address	The IP address of the interface.
OSPF state	ptr2ptr (point to point)
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 = 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 neighbor router's ID.

Encrypted display of the authentication string or MD5 authentication key

The optional 0 | 1 parameter with the **authentication-key** and **md5-authentication key-id** parameters affects encryption.

For added security, BigIron RX encrypts the display of the password or authentication string. Encryption is enabled by default. The software also provides an optional parameter to disable encryption of a password or authentication string, on an individual OSPF area or OSPF interface basis.

When encryption of the passwords or authentication strings is enabled, they are encrypted in the CLI regardless of the access level you are using. The encryption option can be omitted (the default) or can be one of the following:

- **0** – Disables encryption for the password or authentication string you specify with the command. The password or string is shown as clear text in the running configuration and the startup configuration file. Use this option of you do not want display of the password or string to be encrypted.
- **1** – Assumes that the password or authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE
If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

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:

- **Trunk 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.

Changing the reference bandwidth

To change the reference bandwidth, enter a command such as the following at the OSPF configuration level of the CLI:

```
BigIron RX(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*>

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.

```
BigIron RX(config-ospf-router)# no auto-cost reference-bandwidth
```

Define redistribution filters

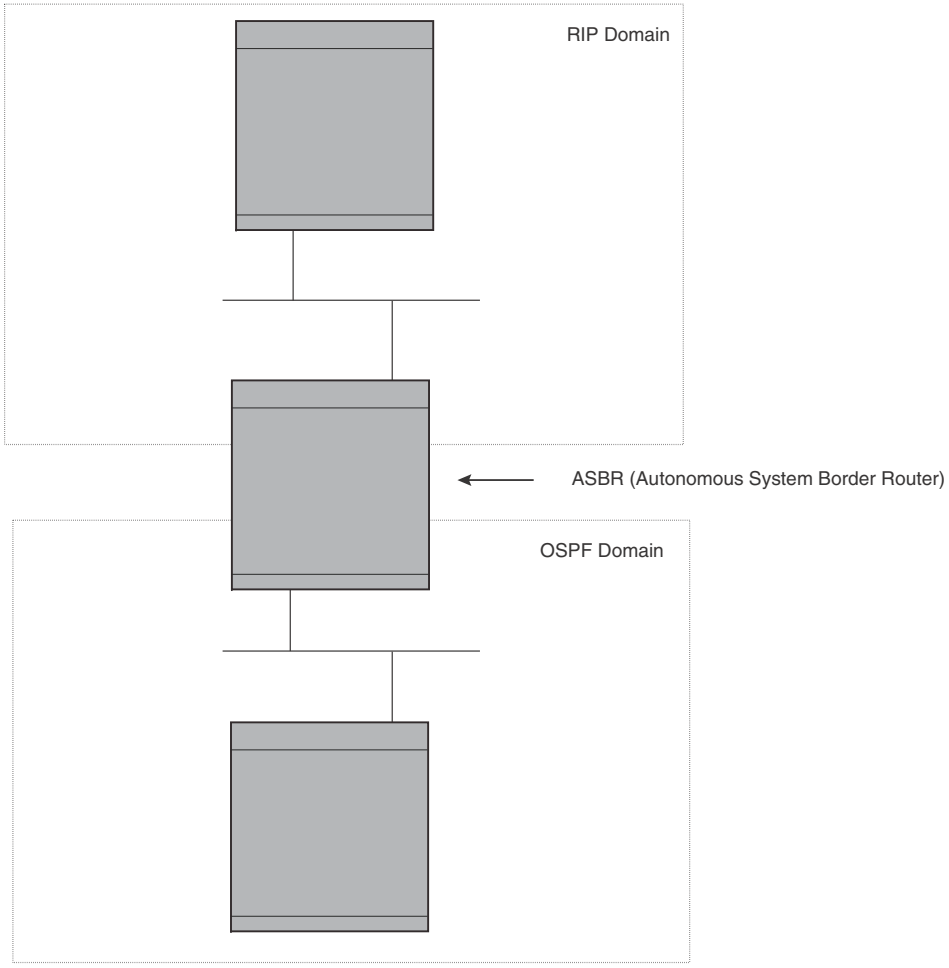
Route redistribution imports and translates different protocol routes into a specified protocol type. On the device, redistribution is supported for static routes, ISIS, OSPF, RIP, and BGP4. OSPF redistribution supports the import of static, ISIS, RIP, and BGP4 routes into OSPF routes.

NOTE
The BigIron RX 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 107 on page 701, 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 107 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] redistribution 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.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)# redistribution rip
BigIron RX(config-ospf-router)# redistribution static
BigIron RX(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.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)# default-metric 4
```

Syntax: default-metric <*value*>

The <*value*> can be from 1 – 65535. 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.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)# redistribution rip
BigIron RX(config-ospf-router)# redistribution static
BigIron RX(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.

```
BigIron RX(config)# ip route 1.1.0.0 255.255.0.0 207.95.7.30
BigIron RX(config)# ip route 1.2.0.0 255.255.0.0 207.95.7.30
BigIron RX(config)# ip route 1.3.0.0 255.255.0.0 207.95.7.30
BigIron RX(config)# ip route 4.1.0.0 255.255.0.0 207.95.6.30
BigIron RX(config)# ip route 4.2.0.0 255.255.0.0 207.95.6.30
BigIron RX(config)# ip route 4.3.0.0 255.255.0.0 207.95.6.30
BigIron RX(config)# ip route 4.4.0.0 255.255.0.0 207.95.6.30 5
BigIron RX(config)# route-map abc permit 1
BigIron RX(config-routemap abc)# match metric 5
BigIron RX(config-routemap abc)# set metric 8
BigIron RX(config-routemap abc)# router ospf
BigIron RX(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.

```
BigIron RX(config-ospf-router)# 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] redistribution 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*>

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.

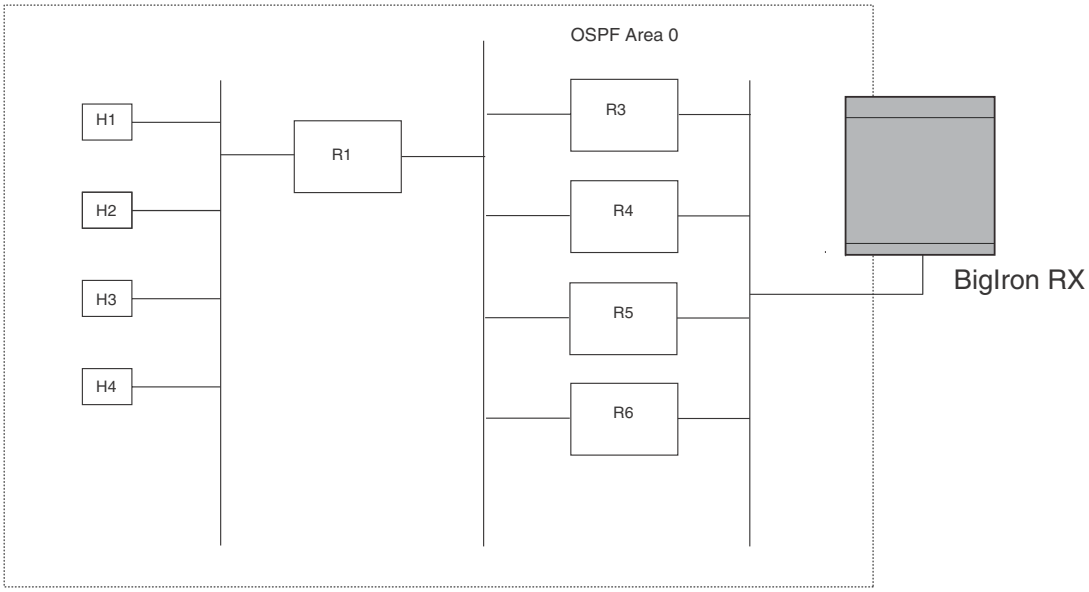
NOTE
For an external route that is redistributed into OSPF through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map. The **default-metric** <*num*> command has no effect on the route. This behavior is different from 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

BigIron RX 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 108 shows an example of an OSPF network containing multiple paths to a destination (in this case, R1).

FIGURE 108 Example OSPF network with four equal-cost paths



In the example in Figure 108, the BigIron RX has four paths to R1:

- BigIron RX ->R3
- BigIron RX ->R4
- BigIron RX ->R5
- BigIron RX ->R6

Normally, the device will choose the path to the R1 with the lower metric. For example, if R3's metric is 1400 and R4's metric 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 BigIron RX is not source routing in these examples. The router 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. To configure IP load sharing parameters, refer to "Configuring IP load sharing" on page 201.

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 BigIron RX 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.

```
BigIron RX(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.

```
BigIron RX(config-ospf-router)# 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
BigIron RX 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 (BigIron RX) will not inject the default route into an NSSA by default and the command described in this section will not cause the BigIron RX 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 685.

To enable default route origination, enter the following command.

```
BigIron RX(config-ospf-router)# default-information-originate
```

To disable the feature, enter the following command.

```
BigIron RX(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:

- **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 you specify are used even if you do not use the **always** option.

Configuring a default network route

The device 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 device 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.

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.

```
BigIron RX(config)# ip default-network 209.157.22.0
BigIron RX(config)# write memory
```

Syntax: 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.

```
BigIron RX(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.

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 ten 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.

```
BigIron RX(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.

Syntax: timers spf <*delay*> <*hold-time*>

The <*delay*> parameter specifies the SPF delay.

The <*hold-time*> parameter specifies the SPF hold time.

To set the timers back to their default values, enter a command such as the following.

```
BigIron RX(config-ospf-router)# no timers spf 10 20
```

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.

```
BigIron RX(config-ospf-router)# metric-type type1
```

Syntax: 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. Refer to "Changing administrative distances" on page 765 for a list of the default distances for all route sources.

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 device's decision 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.

```
BigIron RX(config-ospf-router)# distance external 100
BigIron RX(config-ospf-router)# distance inter-area 90
BigIron RX(config-ospf-router)# distance intra-area 80
```

Syntax: 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.

```
BigIron RX(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'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.

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.

```
BigIron RX(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.

```
BigIron RX(config-ospf-router)# no timers lsa-group-pacing
```

OSPF ABR type 3 LSA filtering

OSPF ABR Type 3 LSA filtering increases the ability of an ABR that is running the OSPF protocol to filter type 3 link-state advertisements (LSAs) that are sent between different OSPF areas. Only packets with specified prefixes will be sent from one area to another area and prohibits all packets with other prefixes.

Type 3 LSAs refer to summary links and are sent by ABRs to advertise destinations outside the area. OSPF ABR Type 3 LSA filtering gives the administrator improved control of route distribution between OSPF areas.

Usage and configuration guidelines

- The "area prefix-list" command is only applicable to the ABRs. If the router is not an ABR the configuration is accepted however it will start working only after router is made ABR.

- With this feature enabled in the “in” direction, all type 3 LSAs originated by the ABR to this area, based on information from all other areas, are filtered by the prefix list. Type 3 LSAs that were originated as a result of the area range command in another area are treated like any other type 3 LSA that was originated individually. Any prefix that does not match an entry in the prefix list is implicitly denied.
- With this feature enabled in the “out” direction, all type 3 LSAs advertised by the ABR, based on information from this area to all other areas, are filtered by the prefix list. If the area range command has been configured for this area, Type 3 LSAs that corresponds to the area range command are treated like any other type 3 LSA.
- Prefixes that are not permitted by the prefix list are implicitly denied.
- The following table displays the behavior for prefix list configurations

TABLE 106 Behavior for prefix list configurations

IP prefix list	OSPF area prefix list	Event	Filtering done
XXX	Not defined	None	No (permit all)
Not defined	Defined	None	Yes (deny all)
Not defined	Defined	IP prefix list defined	Recalculation
Defined (no rules configured)	Defined	None	Implicit deny (deny all)
Defined (rules configured)	Defined	IP prefix list deleted	Recalculation and deny all
Defined (rules configured)	Defined	IP prefix list rule added or modified or deleted	Recalculation
Defined (rules configured)	Defined	Area prefix list deleted	Recalculation and permit all

Configuring an OSPF area prefix list

To filter prefixes advertised in type 3 link-state advertisements (LSAs) between (OSPF) areas of an Area Border Router (ABR), use the area prefix-list command in router configuration mode. To change or cancel the filter, use the no form of this command.

Configuring OSPF ABR type 3 LSA filtering

To filter inter-area routes into a specified area, use the following commands beginning in router configuration mode.

To configure the router to run an OSPF process, enter commands such as the following.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)#
```

To filter prefixes advertised in type 3 link-state advertisements (LSAs) between (OSPF) areas of an Area Border Router (ABR), use the area prefix-list command in router configuration mode. To change or cancel the filter, use the no form of this command.

```
BigIron RX(config-ospf-router)#area 1 prefix-list area 1 in
```

To configure the switch to filter inter-area routes out of the specified area, enter a command such as the following.

```
BigIron RX(config-ospf-router)# area 10.10.10.1 prefix-list Routesfor20 out
```

Syntax: [no] **area** {<*area-id*> | <*area_ip*>} **prefix-list** {prefix-list-name in | out}

The **< prefix-list-name >** parameter specifies the prefix list name.

The **{**<*area-id*> **|** <*area_ip*>**}** parameter specifies the area id in different formats.

The **in** keyword specifies that prefix list is applied to prefixes advertised to the specified area from other areas.

The **out** keyword specifies that prefix list is applied to prefixes advertised out of the specified area to other areas.

Defining and applying IP prefix lists

An IP prefix list specifies a list of networks. When you apply an IP prefix list to an area, 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 an area, enter commands such as the following.

```
BigIron RX(config)# ip prefix-list Routesfor20 permit 20.20.0.0/24
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)#)# area 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 **area** command configures the device to use IP prefix list Routesfor20 to determine which routes to send to area 10.10.10.1. The device sends routes that go to 20.20.x.x to area 10.10.10.1 because the IP prefix list explicitly permits these routes to be sent to the area.

Syntax: 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. You use this name when applying the prefix list to a neighbor.

The **seq** <*seq-value*> parameter is optional and specifies the IP 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 software 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 | permit** parameter specifies the action the software takes if a neighbor's route is in this prefix list.

The prefix-list matches only on this network unless you use the **ge** <*ge-value*> or **le** <*le-value*> parameters. (See below.)

The <*network-addr*>/<*mask-bits*> parameter specifies 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*>.

- If you specify only **ge** <*ge-value*>, then the mask-length range is from <*ge-value*> to 32.
- If you specify only **le** <*le-value*>, then 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 specify with the <*network-addr*>/<*mask-bits*> parameter.

Displaying the configured OSPF area prefix list

To display the prefix-lists attached to the areas, enter the following command.

```
BigIron RX(config)#show ip ospf config
Router OSPF: Enabled
Graceful Restart: Disabled, timer 120
Graceful Restart Helper: Enabled
Redistribution: Disabled
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: 5.5.5.1
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      Prefix List In      Prefix List Out
0             normal  0
1             normal  0         Area_1_Pfx_list in  Area_1_Pfx_List_Out
```

Syntax: show ip ospf config

DIsplaying the configured IP prefix list

To only display the configured ip prefix-list, enter a command such as the following.

```
BigIron RX# show ip prefix-lists
ip prefix-list abc: 2 entries
seq 5 deny 2.3.4.0/24
seq 10 permit 4.5.0.0/16.
```

Syntax: show ip prefix-lists <*prefix-list-name*>

The <*prefix-list-name*> parameter specifies the name of the prefix list. You use this name when applying the prefix list to an area.

Modifing OSPF traps generated

OSPF traps as defined by RFC 1850 are supported on BigIron RX.

You can enable or disable OSPF trap generation by doing the following.

1. Enabling SNMP traps for OSPF. (Refer to "iDisabling and enabling SNMP traps for OSPF" on page 714.)
2. Enable OSPF logging. (Refer to "Enabling OSPF logging" on page 715.)

Refer to Table 107 on page 714 for the list of the default settings for OSPF traps.

TABLE 107 Default settings for OSPF traps

Trap name	default
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

iDisabling and enabling SNMP traps for OSPF

By default, most SNMP trap generation for OSPF is enabled (Refer to Table 107 on page 714 for the OSPF trap default values). You can disable the generation of these traps by entering the following CLI command.

```
BigIron RX(config-ospf-router)# 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 BigIron RX, 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.

```
BigIron RX(config-ospf-router)# no trap neighbor-state-change-trap
```

Example

To reinstate the trap, enter the following command.

```
BigIron RX(config-ospf-router)# trap neighbor-state-change-trap
```

Syntax: [no] snmp-server trap ospf <*ospf-trap*>

Enabling OSPF logging

By default, most OSPF logging is enabled (Refer to Table 107 on page 714 for a complete list of the OSPF default trap settings). If OSPF logging has been previously disabled, you must enable OSPF logging if you want SNMP traps to be generated for OSPF. Enter commands such as the following.

```
BigIron RX(config)#router ospf
BigIron RX(config-ospf-router)#log all
```

Syntax: log all | adjacency | bad_packet | database | memory | retransmit

Enter **all** to log all OSPF traps generated.

Enter **adjacency** to log only the traps for adjacency changes

Enter **bad_packet** to log only those traps for bad packets

Enter **memory** to log only those traps related to memory issues.

Enter **retransmit** to log only those traps related to retransmission activities.

NOTE
OSPF retransmit is not logged when log retransmit option is enabled.

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.

```
BigIron RX(config)# router ospf
BigIron RX(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 BigIron RX 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.

```
BigIron RX(config-ospf-router)# data-base-overflow-interval 60
```

Syntax: 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.

```
BigIron RX(config)# router ospf
BigIron RX(config-ospf-router)# log all
```

Syntax: [no]log all | adjacency | 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 **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.

Displaying OSPF information

You can display the following OSPF information:

- **Trap, area, and interface information** – refer to "Displaying general OSPF configuration information" on page 718.
- **CPU utilization statistics** – refer to "Displaying CPU utilization and other OSPF tasks" on page 719.
- **Area information** – refer to "Displaying OSPF area information" on page 720.
- **Neighbor information** – refer to "Displaying OSPF neighbor information" on page 721.
- **Interface information** – refer to "Displaying OSPF interface information" on page 723.
- **Route information** – refer to "Displaying OSPF route information" on page 725.
- **External link state information** – refer to "Displaying OSPF external link state Information" on page 727.
- **Link state information** – refer to "Displaying OSPF database link state information" on page 728.
- **Virtual Neighbor information** – refer to "Displaying OSPF virtual neighbor and link information" on page 730.
- **Virtual Link information** – refer to "Displaying OSPF virtual link information" on page 732.
- **ABR and ASBR information** – refer to "Displaying OSPF ABR and ASBR information" on page 729.
- **Trap state information** – refer to "Displaying OSPF trap status" on page 730.

Displaying general OSPF configuration information

To display general OSPF configuration information, enter the following command at any CLI level.

```
BigIron RX> 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.

```
BigIron RX#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 108 CLI display of show tasks

This field...	Displays...
Task Name	Name of task running on the BigIron RX.
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.

```
BigIron RX> 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 109 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 109 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 BigIron RX 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.

```
BigIron RX# 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
```

To display OSPF neighbor information by area, enter a command such as the following.

```
BigIron RX# show ip ospf neighbor area 1

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
```

Syntax: show ip ospf neighbor [router-id <*ip-addr*>] | [<*num*>] [area <*ip-addr*> | <*num*>]

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.

These displays show the following information.

TABLE 110 CLI display of OSPF neighbor information

Field	Description
Port	The port through which the BigIron RX is connected to the neighbor.
Address	The IP address of this BigIron RX's interface with 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 110 CLI display of OSPF neighbor information (Continued)

Field	Description
State	The state of the conversation between the BigIron RX 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 Brocade technical support. See 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.

```
BigIron RX# 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 111 Output of the show ip ospf interface command

This field	Displays
IP Address	The IP address of the interface.
OSPF state	ptr2ptr (point to point)
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 = 0x01 • Point to Point = 0x03 • Virtual Link = 0x04

TABLE 111 Output of the **show ip ospf interface** command (Continued)

This field	Displays
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 neighbor router's ID.

Displaying OSPF route information

To display OSPF route information, enter the following command at any CLI level.

```
BigIron RX>#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 112 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 BigIron RX.)
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 BigIron RX.
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 Brocade technical support.
Tag	The external route tag.
Flags	State information for the route entry. This information is used by Brocade technical support.
Paths	The number of paths to the destination.
Out_Port	The router port through which the BigIron RX 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 Brocade 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.

```
BigIron RX# 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.

```
BigIron RX# show ip ospf redistribute route 3.1.0.0 255.255.0.0
  3.1.0.0 255.255.0.0 static
```

Displaying OSPF external link state Information

To display external link state information, enter the following command at any CLI level.

```
BigIron RX>#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)*>] | [status <*num*>]

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 r**outer-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 113 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 BigIron RX learned this route.

TABLE 113 CLI display of OSPF external link state information (Continued)

This field...	Displays...
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 Brocade technical support.

Displaying OSPF database link state information

To display database link state information, enter the following command at any CLI level.

```
BigIron RX> 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] | [extensive] | [link-state-id <*ip-addr*>] | [network] | [nssa] | [router] | [router-id <*ip-addr*>] | [sequence-number <num(Hex)>] | [summary]

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-stat**e command to display the table.

The **asbr** option shows ASBR information.

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 **network** option shows network information.

The **nssa** option shows network information.

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 **summary** option shows summary information.

TABLE 114 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 BigIron RX 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 BigIron RX 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.

```
BigIron RX># 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.

```
BigIron RX#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 115 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

TABLE 115 CLI display of OSPF border routers (Continued)

This field...	Displays...
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 "Modifing OSPF traps generated" on page 714.

To display the state of each OSPF trap, enter the following command at any CLI level.

```
BigIron RX># 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

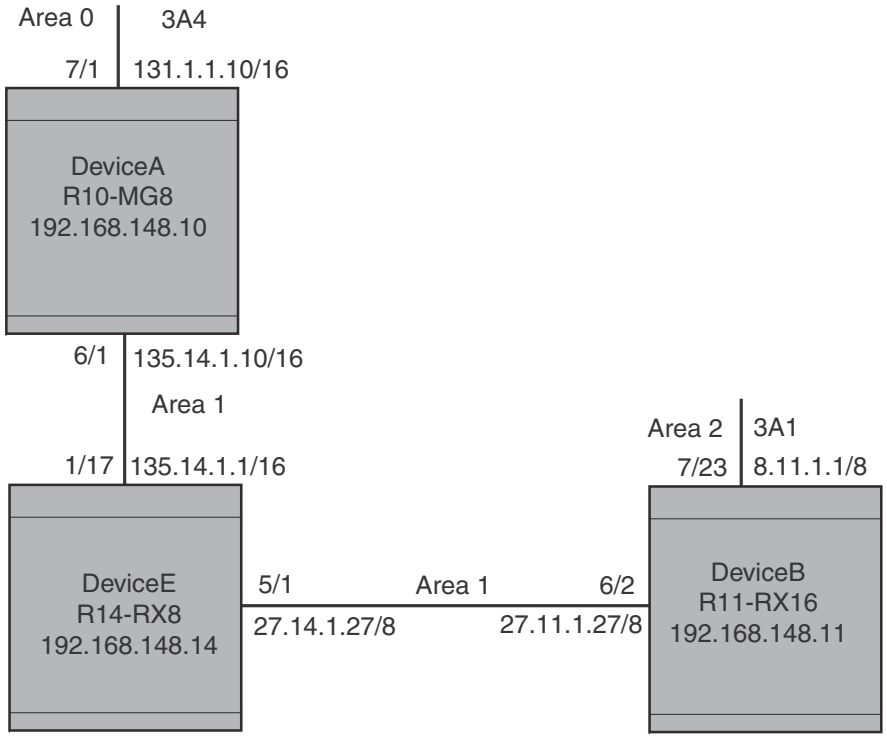
Displaying OSPF virtual neighbor and link information

You can display OSPF virtual neighbor and virtual link information. For example, the following show run display shows the configuration in Figure 109.

```
BigIron RX#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 109 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 109.

```
BigIron RX#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 109.

```
BigIron RX#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.

OSPF graceful restart

With OSPF graceful restart enabled, a restarting router sends special LSAs, called grace-lsas, to its neighbors. These LSAs are sent to neighbors either before a planned OSPF restart or immediately after an unplanned restart. A grace LSA contains a grace period value that the requesting routers asks its neighbor routers to use for the existing routes, to and through the router after a restart. The restarting router comes up, it continues to use its existing OSPF routes to forward packets. In the background, it re-establishes OSPF adjacencies with its neighboring router, relearns all OPSF LSAs, recalculates its OSPF routes, and replaces them with new routes as necessary. Once the restarting router relearns all OSPF routes, it flushes the grace LSAs from the network, informing the helper routers of the completion of the restart process. If the restarting router does not re-establish adjacencies with the helper router within the restart time, the helper router stops the helping function and flushes the stale OSPF routes.

Configuring OSPF graceful restart

To configure OSPF Graceful Restart on a router, the restarting router and its directly connected OSPF peers must be configured with Graceful Restart.

```
BigIron RX(config)#router ospf
BigIron RX(config-ospf-router)#area 0
BigIron RX(config-ospf-router)#graceful-restart
```

graceful-restart

Enabling and disabling OSPF helper

When OSPF is enabled, the helper mode is enabled by default. OSPF routers that do not have graceful restart enabled will act as if the graceful restart helper is enabled. To prevent the graceful restart from performing its function, disable it by entering the following command.

```
BigIron RX(config-ospf-router)#graceful-restart helper-disable
```

Syntax: [no] graceful-restart helper disable

Use the **no** form of the command to re-enable the graceful restart helper.

Configuring OSPF graceful restart timer

The OSPF graceful restart timer specifies the maximum amount of time an OSPF restarting router will take to re-establish OSPF adjacencies and relearn OSPF routes. This value will be sent to the neighboring routers in the grace LSA packets. Configure the timer by entering a command such as the following.

```
BigIron RX(config-ospf-router)#graceful-restart restart-time 120
```

Syntax: graceful-restart restart-time <*seconds*>

Enter 10 – 1800 for seconds. The default is 120.

Displaying OSPF graceful restart information

Displaying if OSPF graceful restart is enabled

Use the **show ip ospf data grace-link-state** and the **show ip ospf neighbor** commands to display information about OSPF graceful restart.

The following is an example of what the **show ip ospf data grace-link-state** command that is displayed during a restart event. The output is blank if the report is requested while the OSPF router is in normal operation.

```
BigIron RX# show ip ospf data grace-link-state
    Area  Interface  Router ID    Type  Age      Restart-Time  Seq
     0    3/27       12.1.0.14    9     27        120          0x80000001
     0     v31       12.1.0.14    9     27        120          0x80000001
     0     v32       12.1.0.14    9     27        120          0x80000001
     0     v33       12.1.0.14    9     27        120          0x80000001
     0     v34       12.1.0.14    9     27        120          0x80000001
```

The **show ip ospf neighbor** command displays the following information during normal operation.

```
BigIron RX# show ip ospf neighbor
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/1   30.1.0.5        0   FULL/OTHER 30.1.0.13       30.0.0.13       5  2   0
3/27  25.27.0.8       1   FULL/DR    25.27.0.14      12.1.0.14       20 2   0
v31   21.23.0.5       1   FULL/DR    21.23.0.14      12.1.0.14       15 2   0
v32   22.24.0.5       1   FULL/DR    22.24.0.14      12.1.0.14       15 2   0
v33   23.25.0.5       1   FULL/DR    23.25.0.14      12.1.0.14       15 2   0
v34   24.26.0.5       1   FULL/DR    24.26.0.14      12.1.0.14       15 2   0
```

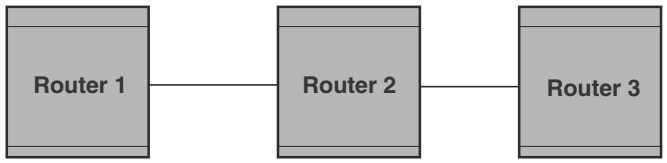
The **show ip ospf neighbor** command displays the following information during a restart event on a helper router. Note the "<in graceful restart state...>" entry appears only during restart. It does not appear once restart is complete.

```
BigIron RX#sh ip ospf neigh
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/1   30.1.0.5        0   FULL/OTHER 30.1.0.13       30.0.0.13       5  2   0
3/27  25.27.0.8       1   FULL/DR    25.27.0.14      12.1.0.14       20 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v31   21.23.0.5       1   FULL/DR    21.23.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v32   22.24.0.5       1   FULL/DR    22.24.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v33   23.25.0.5       1   FULL/DR    23.25.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v34   24.26.0.5       1   FULL/DR    24.26.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
```

Example

OSPF Graceful Restart requires at least three routers as shown in Figure 110.

FIGURE 110 Restarting router topology



Restarting Router

Before configuring graceful restart, use the **show ip ospf neighbor** command to determine the state of the OSPF neighbors. For example:

```
BigIron RX# show ip ospf neighbor
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/7   40.0.1.1        1   FULL/DR    40.0.1.3        9.0.1.24        23 2   0
```

Enable graceful restart on each OSPF router in Figure 110. For example,

Router 1

```
BigIron RX(config)#router ospf
BigIron RX(config-ospf-router)#graceful-restart
BigIron RX(config-ospf-router)#area 0
```

Router 2

```
BigIron RX(config)#router ospf
BigIron RX(config-ospf-router)#graceful-restart
BigIron RX(config-ospf-router)#area 0
```

Router 3

```
BigIron RX(config)#router ospf
BigIron RX(config-ospf-router)#graceful-restart
BigIron RX(config-ospf-router)#area 0
```

Use the **show ip ospf neighbor** command to display the state of the OSPF neighbors after enabling graceful restart. For example:

```
BigIron RX 1# show ip ospf neigh
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/7   40.0.1.1        1   EXST/DR    40.0.1.3        9.0.1.24        24 2   0
  < in graceful restart state,  helping 1, timer 112 sec >

BigIron RX 3# show ip ospf neighbor
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
2/2   40.0.10.1       1   EXST/DR    40.0.10.3       8.0.0.23        23 2   0
< in graceful restart state,  helping 1, timer 111 sec >
```

Note the "<in graceful restart state...>" entry appears only during restart. It does not appear once restart is complete. The restarting router should resync LSDB with its peers when the restart has completed.

Chapter

26

Configuring BGP4 (IPv4 and IPv6)

In this chapter

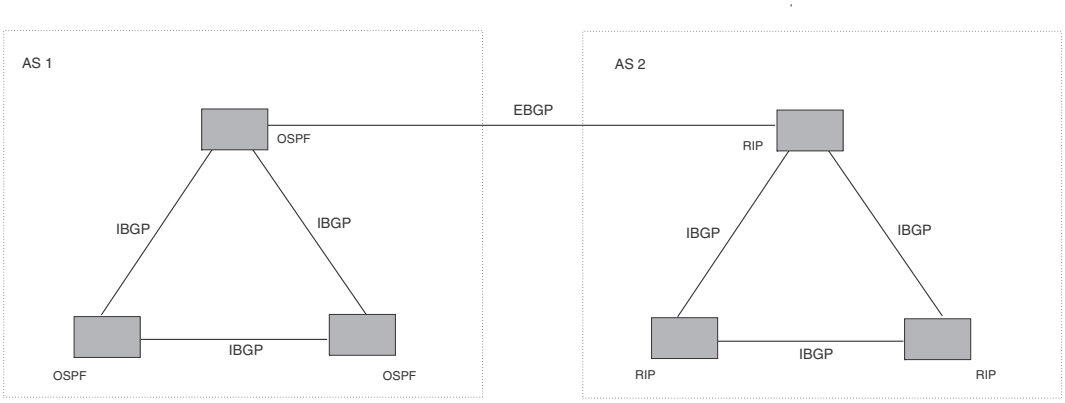
Overview of BGP4

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.

Routers within an AS can use different Interior Gateway Protocols (IGPs) such as RIP and OSPF to communicate with one another. However, for routers in different ASs to communicate, they need to use an EGP. BGP4 is the standard EGP used by Internet routers and therefore is the EGP implemented on BigIron RX.

Figure 111 on page 739 shows a simple example of two BGP4 ASs. Each AS contains three BGP4 routers. All of the BGP4 routers within an AS communicate using IBGP. BGP4 routers communicate with other ASs using EBGP. Notice that each of the routers also is running an Interior Gateway Protocol (IGP). The routers in AS1 are running OSPF and the routers 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 111 Example BGP4 ASs



Relationship between the BGP4 route table and the IP route table

The BigIron RX's 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 router that also is running BGP4. BGP4 neighbors communicate using Transmission Control Protocol (TCP) port 179 for BGP communication. When you configure the device for BGP4, one of the configuration tasks you perform is to identify the BigIron RX's BGP4 neighbors.

Although a router'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 BGP chooses and sends to the IP route table is the ***preferred route***. This route is what the device advertises to other BGP 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 BigIron RX advertises a route to one of its neighbors, the route is expressed in this format.
- **AS-path** – A list of the other ASs through which a route passes. BGP4 routers can use the AS-path to detect and eliminate routing loops. For example, if a route received by a BGP4 router contains the AS that the router is in, the router does not add the route to its own BGP4 table. (The BGP4 RFCs refer to the AS-path as "AS_PATH".)
- **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 BigIron RX re-advertises a learned best BGP4 route to the BigIron RX'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 BGP selects based on comparison of the BGP4 route path's attributes.

After a BigIron RX 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 routers send UPDATE messages to inform neighbors of new, changed, or no longer feasible routes. BGP4 routers do not send regular updates. However, if configured to do so, a BGP4 router does regularly send KEEPALIVE messages to its peers to maintain BGP4 sessions with them if the router does not have any route information to send in an UPDATE message. Refer to "BGP4 message types" on page 741 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 router, the router 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 path.

   NOTE
   By default, the device does not use the default route to resolve BGP4 next hop. Also refer to "Enabling next-hop recursion" on page 781 and "Using the IP default route as a valid next hop for a BGP4 route" on page 781

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 BGP4 BigIron RX).
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 **bgp-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.

- BigIron RX 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.

  In addition, you can 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 BigIron RX favoring the route paths that are missing their MEDs. You can use the **med-missing-as-worst** command to make the BigIron RX regard a BGP 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 router 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
BigIron RX supports BGP4 load sharing among multiple equal-cost paths. BGP4 load sharing enables the BigIron RX to balance the traffic across the multiple paths instead of choosing just one path based on router 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. Prefer the path that comes from the BGP4 router with the lowest router ID, if **compare-router ID** is enabled. If a path contains originator ID attributes, then originator ID is substituted for the ROUTER ID in the decision process.
12. Prefer the path with the minimum cluster list length.
13. If the route is a BGP VRF instance, prefer the route with the smallest RD value.

BGP4 message types

BGP4 routers communicate with their neighbors (other BGP4 routers) using the following types of messages:

- OPEN
- UPDATE

- KEEPALIVE
- NOTIFICATION
- ROUTE REFRESH

OPEN message

After a BGP4 router establishes a TCP connection with a neighboring BGP4 router, the routers exchange OPEN messages. An OPEN message indicates the following:

- **BGP version** – Indicates the version of the protocol that is in use on the router. BGP 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** – A two-byte number that identifies the AS to which the BGP4 router belongs.
- **Hold Time** – The number of seconds a BGP4 router will wait for an UPDATE or KEEPALIVE message (described below) from a BGP4 neighbor before assuming that the neighbor is dead. BGP4 routers 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 router closes its TCP connection to the neighbor and clears any information it has learned from the neighbor and cached.

  You can configure the Hold Time to be 0, in which case a BGP4 router will consider its 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 fallover feature, which is disabled by default.
- **BGP Identifier** – The router ID. The BGP Identifier (router ID) identifies the BGP4 router to other BGP4 routers. The device use the same router ID for OSPF and BGP4. If you do not set a router ID, the software uses the IP address on the lowest numbered loopback interface configured on the router. If the device does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, refer to "Changing the router ID" on page 788.
- **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 its 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 path information, route preference, next hop values, and aggregation information. BGP4 uses the path attributes to make filtering and routing decisions.
- **Unreachable routes** – A list of routes that have been in the sending router's 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 routers do not regularly exchange UPDATE messages to maintain the BGP4 sessions. For example, if a BigIron RX configured to perform BGP4 routing has already sent the latest route information to its peers in UPDATE messages, the router does not send more UPDATE messages. Instead, BGP4 routers send KEEPALIVE messages to maintain the BGP4 sessions. KEEPALIVE messages are 19 bytes long and consist only of a message header; they contain no routing data.

BGP4 routers send KEEPALIVE messages at a regular interval, the Keep Alive Time. The default Keep Alive Time on BigIron RX is 60 seconds.

A parameter related to the Keep Alive Time is the Hold Time. A BGP4 router's Hold Time determines how many seconds the router will wait for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead. The Hold Time is negotiated when BGP4 routers exchange OPEN messages; the lower Hold Time is then used by both neighbors. For example, if BGP4 Router A sends a Hold Time of 5 seconds and BGP4 Router B sends a Hold Time of 4 seconds, both routers 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 router assumes that its neighbor is alive regardless of how many seconds pass between receipt of UPDATE or KEEPALIVE messages.

NOTIFICATION message

When you close the router's BGP4 session with a neighbor, or the router detects an error in a message received from the neighbor, or an error occurs on the router, the router sends a NOTIFICATION message to the neighbor. No further communication takes place between the BGP4 router that sent the NOTIFICATION and the neighbors that received the NOTIFICATION.

REFRESH message

BGP sends a REFRESH message to a neighbor to request the neighbor to resend route updates. This type of message can be useful if an inbound route filtering policy has been changed.

Brocade implementation of BGP4

BGP4 is described in RFC 1771 and the latest BGP drafts. The Brocade implementation fully complies with RFC 1771 and also supports the following:

- RFC 1745 (OSPF Interactions)
- RFC 1997 (BGP Communities Attributes)
- RFC 2385 (TCP MD5 Signature Option)

- RFC 2439 (Route Flap Dampening)
- RFC 2796 (Route Reflection)
- RFC 2842 and 3392 (Capability Advertisement)
- RFC 3065 (BGP4 Confederations)
- RFC 2858 (Multiprotocol Extensions)
- RFC 2918 (Route Refresh Capability)
- RFC 3392 (BGP Capability Advertisement)

Memory considerations

BGP4 handles a very large number of routes and therefore requires a lot of memory. For example, in a typical configuration with just a single BGP4 neighbor, a BGP4 router may need to be able to hold up to 150,000 routes. Many configurations, especially those involving more than one neighbor, can require the router to hold even more routes. The device provide dynamic memory allocation for BGP4 data. These 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, BigIron RX switches 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 around two million.

Configuring BGP4

Once you activate BGP, you can configure the BGP options. On a BigIron RX there are two configuration levels: global and address family.

At the *global level,* all BGP configurations apply to IPv4 and IPv6. You enter this layer using the **router bgp** command.

Under the global level, you specify an ***address family***. Address families separate the IPv4 and IPv6 BGP configuration. You enter this level by entering the **address-family** command at the router bgp 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. You specify multicast or unicast routes.

FIGURE 112 BGP configuration levels

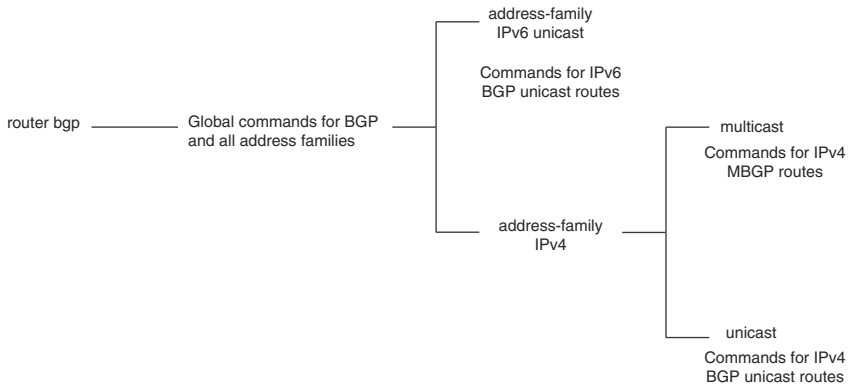


Table 26.1 shows what commands are available at the various BGP configuration levels.

TABLE 116 IPv4 BGP commands at 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 749
address-filter		x		"Filtering specific IP addresses" on page 749
aggregate-address		x	x	"Aggregating routes advertised to BGP4 neighbors" on page 757
always-compare-med	x			"Configuring the BigIron RX to always compare Multi-Exit Discriminators (MEDs)" on page 757
as-path-filter		x		
as-path-ignore	x			"Disabling or re-enabling comparison of the AS-path length" on page 758
bgp-redistribute-internal		x		"Redistributing IBGP routes" on page 758
client-to-client-reflection				"Disabling or re-enabling client-to-client route reflection" on page 759
cluster-id	x			"Configuring a route reflector" on page 759
community-filter		x		
compare-routerid	x			"Enabling or disabling comparison of the router IDs" on page 759
confederation	x			"Configuring confederations" on page 760
dampening		x	x	"Configuring route flap dampening" on page 763
default-information-originate		x	x	"Originating the default route" on page 764

TABLE 116 IPv4 BGP commands at different configuration levels (Continued)

Command	Global (iPv4 and IPv6)	IPv4 address family unicast	IPv4 address family multicast	See
default-local-preference	x			"Changing the default local preference" on page 764
default-metric		x	x	"Changing the default metric used for redistribution" on page 765
distance	x			"Changing administrative distances" on page 765
enforce-first-as	x			"Requiring the first AS to be the neighbor's AS" on page 766
exit-address-family	x	x	x	"Entering and exiting the address family configuration level" on page 749
fast-external-fallover	x			"Enabling fast external fallover" on page 767
local-as	x			"Setting the local AS number" on page 767
maximum-paths		x		"Changing the maximum number of shared BGP4 paths" on page 768
med-missing-as-worst	x			"Treating missing MEDs as the worst MEDs" on page 768
multipath		x		"Customizing BGP4 load sharing" on page 769
neighbor	x	x	x	"Configuring BGP4 neighbors" on page 769 "Configuring a BGP4 peer group" on page 776
network		x	x	"Specifying a list of networks to advertise" on page 779
next-hop-enable-default		x		"Using the IP default route as a valid next hop for a BGP4 route" on page 781
next-hop-recursion		x		"Enabling next-hop recursion" on page 781
redistribute		x	x	"Modifying redistribution parameters" on page 784
show	x	x	x	"Displaying BGP4 information" on page 822
table-map		x	x	"Using a table map to set the tag value" on page 787
timers	x			"Changing the keep alive time and hold time" on page 787
update-time		x	x	"Changing the BGP4 next-hop update timer" on page 788

When parameter changes take effect

Some parameter changes take effect immediately while others do not take full effect until the router's sessions with its neighbors are reset.

Immediately

The following parameter changes take effect immediately:

- Enable or disable BGP.
- 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's AS.
- Change MED comparison parameters.
- Disable comparison of the AS-Path length.
- Enable comparison of the router 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.

After resetting neighbor sessions

The following parameter changes take effect only after the router's BGP4 sessions are cleared, or reset using the "soft" clear option (Refer to "Closing or resetting a neighbor session" on page 820):

- Change the Hold Time or Keep Alive Time.
- Add, change, or negate filter tables that affect inbound and outbound route policies.

After disabling and re-enabling redistribution

The following parameter change takes effect only after you disable and then re-enable redistribution:

- Change the default MED (metric).

Activating and disabling BGP4

BGP4 is disabled by default. To enable BGP4 and place your BigIron RX into service as a BGP4 router, you must perform the following required 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 router) 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.

```
BigIron RX> enable
BigIron RX# configure terminal
BigIron RX(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
BigIron RX(config-bgp)# local-as 10
BigIron RX(config-bgp)# write memory
```

The **router bgp** command enables the BGP4 protocol.

(For information on the local AS number, refer to "Setting the local AS number" on page 767.)

NOTE
By default, the Brocade router ID is the IP address configured on the lowest numbered loopback interface. If the BigIron RX does not have a loopback interface, the default router ID is the lowest numbered IP interface address configured on the device. For more information, refer to "Changing the router ID" on page 788. If you change the router ID, all current BGP4 sessions are cleared.

NOTE
When BGP4 is enabled on a BigIron RX, you do not need to reset the system. The protocol is activated as soon as you enable it. Moreover, the router begins a BGP4 session with a BGP4 neighbor as soon as you add the neighbor.

Note regarding 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 configuration from the startup configuration. Moreover, when you save the configuration to the startup configuration file after disabling the protocol, 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.

```
BigIron RX(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 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.

To disable BGP4 without losing the BGP4 configuration information, remove the local AS (for example, by entering the **no local-as** <*num*> command). In this case, BGP4 retains the other configuration information but is not operational until you set the local AS again.

Entering and exiting the address family configuration level

The BGP address family has a unicast or multicast sub-level.

To enter the IPv4 BGP unicast address family configuration level, enter the following command.

```
BigIron RX(config-bgp)# address-family ipv4 unicast
BigIron RX(config-bgp)#
```

NOTE
The CLI prompt for the global BGP level and the BGP address-family IPv4 unicast level are the same.

To enter the IPv4 BGP multicast address family configuration level, enter the following command.

```
BigIron RX(config-bgp)# address-family ipv4 multicast
BigIron RX(config-bgp-ipv4m)#
```

Syntax: [no] address-family ipv4 unicast | ipv4 multicast

The default is the ipv4 unicast address family level.

To exit an address family configuration level, enter the following command.

```
BigIron RX(config-bgp-ipv6u)# exit-address-family
BigIron RX(config-bgp)#
```

Syntax: exit-address-family

Filtering specific IP addresses

You can configure the router to explicitly permit or deny specific IP addresses received in updates from BGP4 neighbors by defining IP address filters. The router permits all IP addresses by default. You can define up to 100 IP address filters for BGP4.

- If you want permit to remain the default behavior, define individual filters to deny specific IP addresses.
- If you want to change the default behavior to deny, define individual filters to permit specific IP addresses.

NOTE
Once you define a filter, the default action for addresses that do not match a filter is “deny”. To change the default action to “permit”, configure the last filter as “permit any any”.

Address filters can be referred to by a BGP neighbor's distribute list number as well as by match statements in a route map.

NOTE
If the filter is referred to by a route map’s match statement, the filter is applied in the order in which the filter is listed in the match statement.

NOTE
You also can filter on IP addresses by using IP ACLs. See “Software-Based IP Access Control Lists (ACLs)”.

To define an IP address filter to deny routes to 209.157.0.0, enter the following command.

```
BigIron RX(config-bgp)# address-filter 1 deny 209.157.0.0 255.255.0.0
```

Syntax: [no] address-filter <*num*> permit | deny <*ip-addr*> <*wildcard*> <*mask*> <*wildcard*>

The <*num*> parameter is the filter number.

The **permit | deny** parameter indicates the action the device takes if the filter match is true.

- If you specify **permit**, the BigIron RX permits the route into the BGP4 table if the filter match is true.
- If you specify **deny**, the BigIron RX denies the route from entering the BGP4 table if the filter match is true.

NOTE
Once you define a filter, the default action for addresses that do not match a filter is “deny”. To change the default action to “permit”, configure the last filter as “permit any any”.

The <*ip-addr*> parameter specifies the IP address. If you want the filter to match on all addresses, enter **any**.

The <*wildcard*> parameter specifies the portion of the IP 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
<*ip-addr*> 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 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 configuration 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 configuration 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. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the filter regardless of whether the software is configured to display the masks in CIDR format.

The <*mask*> parameter specifies the network mask. If you want the filter to match on all destination addresses, enter **any.** The wildcard works the same as described above.

Defining an AS-path filter

To define an AS-path filter, enter the command such as the following.

```
BigIron RX(config-bgp)# as-path-filter 4 permit 2500
```

The command defines AS-path filter 4 to permit AS 2500.

Syntax: [no] as-path-filter <*num*> permit | deny <*as-path*>

The <*num*> parameter identifies the filter's position in the AS-path filter list and can be from 1 – 100. Thus, the AS-path filter list can contain up to 100 filters. The device applies the filters in numerical order, beginning with the lowest-numbered filter. When a filter match is true, the device stops and does not continue applying filters from the list.

NOTE
If the filter is referred to by a route map's match statement, the filter is applied in the order in which the filter is listed in the match statement.

The **permit | deny** parameter indicates the action the router takes if the filter match is true.

- If you specify **permit**, the router permits the route into the BGP4 table if the filter match is true.
- If you specify **deny**, the router denies the route from entering the BGP4 table if the filter match is true.

The <*as-path*> parameter indicates the AS-path information. You can enter an exact AS-path string if you want to filter for a specific value. You also can use regular expressions in the filter string.

Defining a community filter

To define filter 3 to permit routes that have the NO_ADVERTISE community, enter the following command.

```
BigIron RX(config-bgp)# community-filter 3 permit no-advertise
```

Syntax: [no] community-filter <*num*> permit | deny <*num*>:<*num*> | internet | local-as | no-advertise | no-export

The <*num*> parameter identifies the filter's position in the community filter list and can be from 1 – 100. Thus, the community filter list can contain up to 100 filters. The router applies the filters in numerical order, beginning with the lowest-numbered filter. When a filter match is true, the router stops and does not continue applying filters from the list.

NOTE
If the filter is referred to by a route map's match statement, the filter is applied in the order in which the filter is listed in the match statement.

The **permit | deny** parameter indicates the action the router takes if the filter match is true.

- If you specify **permit**, the router permits the route into the BGP4 table if the filter match is true.
- If you specify **deny,** the router denies the route from entering the BGP4 table if the filter match is true.

The <*num*>:<*num*> parameter indicates a specific community number to filter. Use this parameter to filter for a private (administrator-defined) community. You can enter up to 20 community numbers with the same command.

If you want to filter for the well-known communities "LOCAL_AS", "NO_EXPORT" or "NO_ADVERTISE", use the corresponding keyword (described below).

The **internet** keyword checks for routes that do not have the community attribute. Routes without a specific community are considered by default to be members of the largest community, the Internet.

The ***local-as*** keyword checks for routes with the well-known community "LOCAL_AS". This community applies only to confederations. The device advertises the route only within the sub-AS. For information about confederations, refer to "Configuring confederations" on page 760.

The **no-advertise** keyword filters for routes with the well-known community "NO_ADVERTISE". A route in this community should not be advertised to any BGP4 neighbors.

The **no-export** keyword filters for routes with the well-known community "NO_EXPORT". A route in this community should not be advertised to any BGP4 neighbors outside the local AS. If the router is a member of a confederation, the device advertises the route only within the confederation. For information about confederations, refer to "Configuring confederations" on page 760.

Configuring a switch to allow routes with its own AS number

BGP rejects routes that contain its own AS number within its AS_PATH attribute to prevent routing loops. In an VPN hub and spoke topology this can stop legitimate routes from being accepted. In this release, the **allowas-in** command eliminates 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 switch to disable the AS_PATH check function for routes sent to it by its BGP neighbor for a maximum limit of 3 occurrences of the route, enter the following command at the BGP configuration level.

```
BigIron RX(config-bgp-ipv4u)# neighbor 33.33.36.2 allowas-in 3
```

Syntax: neighbor <*IPaddress*> allowas-in <*asn_limit*>

The <*IPaddress*> variable is the IP address of the neighbor.

The asn_limit value prevents loops by limiting the number of occurrences that the AS number will be accepted in routes that are received from the specified switch. The maximum limit is 10.

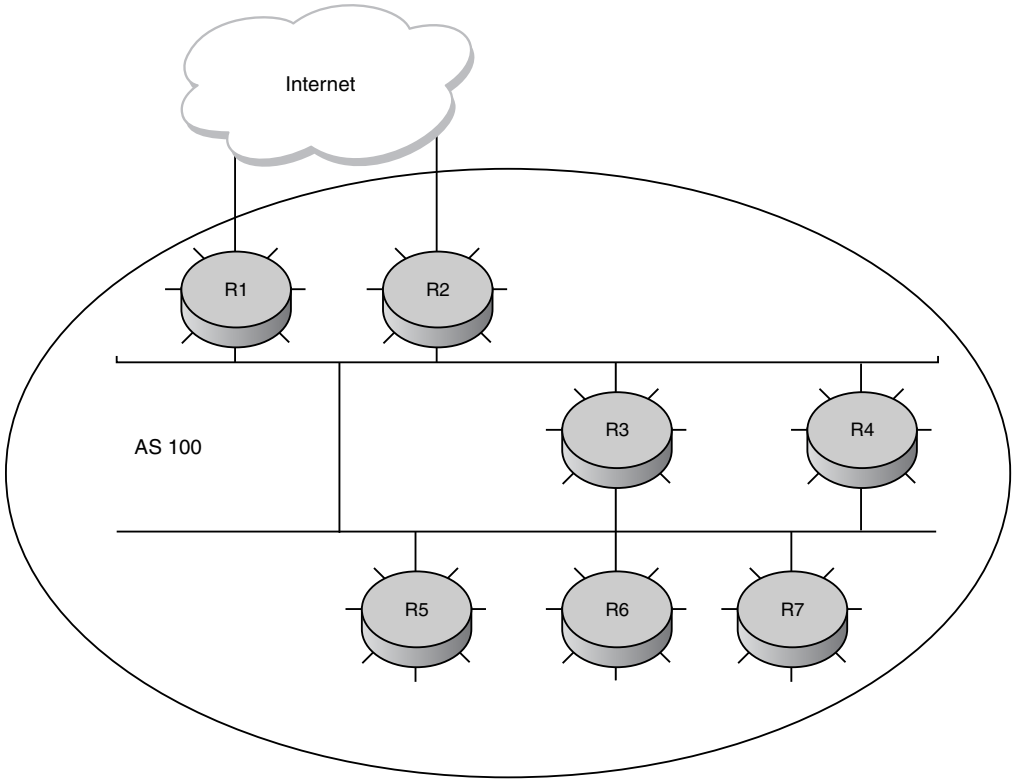
BGP Null0 routing

BGP can use the null0 route to resolve its next hop. Thus, null0 route in the routing table (for example, static route) is considered as a valid route by BGP. If the next hop for BGP resolves into a null0 route, the BGP route is also installed as a null0 route in the routing table.

The null0 routing feature allows network administrators to block certain network prefixes, by using null0 routes and route-maps. The combined use of null0 routes and route maps blocks traffic from a particular network prefix, telling a remote router to drop all traffic for this network prefix by redistributing a null0 route into BGP.

Figure 113 shows a topology for a null0 routing application example.

FIGURE 113 Sample Null0 routing application



The following steps configure a null0 routing application for stopping denial of service attacks from remote hosts on the internet.

Configuration steps

1. Select one router, Router 6, to distribute null0 routes throughout the BGP network.
2. Configure a route-map to match a particular tag (50) and set the next-hop address to an unused network address (199.199.1.1).
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 Router 6, redistribute the static routes into BGP, using route-map <route-map-name> (redistribute static route-map block user).
6. On Router 1, the router facing the internet, configure a null0 route matching the next-hop address in the route-map (ip route 199.199.1.1/32 null0).
7. Repeat step 3 for all routers interfacing with the internet (edge corporate routers). In this case, Router 2 has the same null0 route as Router 1.
8. On Router 6, configure the network prefixes associated with the traffic you want to drop. The static route IP address references a destination address. You are required to 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

Router 6

The following configuration defines specific prefixes to filter.

```
BigIron RX(config)#ip route 110.0.0.40/29 ethernet 3/7 tag 50
BigIron RX(config)#ip route 115.0.0.192/27 ethernet 3/7 tag 50
BigIron RX(config)#ip route 120.014.0/23 ethernet 3/7 tag 50
```

The following configuration redistributes routes into BGP.

```
BigIron RX(config)#router bgp
BigIron RX(config-bgp-router)#local-as 100
BigIron RX(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#redistribute static route-map blockuser
BigIron RX(config-bgp-router)#exit
```

The following configuration defines the specific next hop address and sets the local preference to preferred.

```
BigIron RX(config)#route-map blockuser permit 10
BigIron RX(config-routemap blockuser)#match tag 50
BigIron RX(config-routemap blockuser)#set ip next-hop 199.199.1.1
BigIron RX(config-routemap blockuser)#set local-preference 1000000
BigIron RX(config-routemap blockuser)#set origin igp
BigIron RX(config-routemap blockuser)#exit
```

Router 1

The following configuration defines the null0 route to the specific next hop address. The next hop address 199.199.1.1 points to 128.178.1.101, which gets blocked.

```
BigIron RX(config)# ip route 199.199.1.1/32 null0
BigIron RX(config)#router bgp
local-as 100
BigIron RX(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

Router 2

The following configuration defines a null0 route to the specific next hop address. The next hop address 199.199.1.1 points to 128.178.1.101, which gets blocked.

```
BigIron RX(config)#ip route 199.199.1.1/32 null0
BigIron RX(config)#router bgp
BigIron RX(config-bgp-router)#local-as 100
BigIron RX(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
BigIron RX(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

After configuring the null0 application, you can display the configuration using the **show ip route static, show ip bgp route**, and **show ip route** commands.

For example, when you issue the **show ip route static** command on Router 6, you see the following output.

```
BigIron RX# 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
BigIron RX#
```

Entering a **show ip route static** on Router 1 and Router 2 displays the following.

```
BigIron RX# 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
BigIron RX#
```

Entering a **show BGP route** on Router 6 displays its routing table.

```
Router-6# 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: ..                   .                      .           .     .
```

Issuing a **show ip route** on Router 1 and Router 2 shows “drop” under the Port column for the network prefixes you configured with null0 routing.

```
BigIron RX# 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
BigIron RX#
```

Aggregating routes advertised to BGP4 neighbors

By default, the device advertises individual routes for all the networks. The aggregation feature allows you to configure the device to aggregate routes in 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 instead send a single, aggregate route for the networks. 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.

```
BigIron RX(config-bgp)# aggregate-address 209.157.0.0 255.255.0.0
```

Syntax: 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 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 router to advertise the more specific routes in the specified route map.

The **attribute-map** <*map-name*> parameter configures the router 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 798 for information on defining a route map.

Configuring the BigIron RX to always compare Multi-Exit Discriminators (MEDs)

A Multi-Exit Discriminator (MED) is a value that the BGP4 algorithm uses when comparing multiple paths received from different BGP4 neighbors in the same AS for the same route. In BGP4, a route's MED 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.

In addition, you can 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.

You can enable the device to always compare the MEDs, regardless of the AS information in the paths. For example, if the router receives UPDATES for the same route from neighbors in three ASs, the router would compare the MEDs of all the paths together, rather than comparing the MEDs for the paths in each AS individually.

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 BigIron RX favoring the route paths that are missing their MEDs. You can use the **med-missing-as-worst** command to make the BigIron RX regard a BGP route with a missing MED attribute as the least favorable route, when comparing the MEDs of the routes.

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.

NOTE
The AS-path is empty when routes are redistributed from other protocols (e.g. redistribute static, redistribute connected, or redistribute OSPF).

To configure the router to always compare MEDs, enter the following command.

```
BigIron RX(config-bgp)# always-compare-med
```

Syntax: [no] always-compare-med

Release 02.4.01 of the Multi-Service IronWare software initiated support for the following new BGP command that directs BGP to take the MED value into consideration even if the route has an empty as-path path attribute.

```
BigIron RX(config) router bgp
BigIron RX(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 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 BGP configuration level of the CLI.

```
BigIron RX(config-bgp)# 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 26-3 skips from Step 4 to Step 6.

Syntax: [no] as-path-ignore

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 the 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.

```
BigIron RX(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.

```
BigIron RX(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; the 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 following command. When the feature is disabled, route reflection does not occur between clients but reflection does still occur between clients and non-clients.

```
BigIron RX(config-bgp)# no client-to-client-reflection
```

Enter the following command to re-enable the feature.

```
BigIron RX(config-bgp)# client-to-client-reflection
```

Syntax: [no] client-to-client-reflection

Configuring a route reflector

You can configure one cluster ID on the router. All route-reflector clients for the router are members of the cluster.

To configure a BigIron RX as route reflector 1, enter the following command.

```
BigIron RX(config-bgp)# cluster-id 1
```

Syntax: [no] cluster-id <*num*> | <*ip-addr*>

The <*num*> | <*ip-addr*> parameter specifies the cluster ID (1 – 4294967295) or an IP address. The default is the router 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 the router IDs

Router ID comparison is step 11 in the algorithm BGP4 uses to select the next path for a route.

NOTE
Comparison of router IDs is applicable only when BGP4 load sharing is disabled.

When router ID comparison is enabled, the path comparison algorithm compares the router 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 router ID.
- If BGP4 load sharing is enabled, the device load shares among the remaining paths. In this case, the router ID is not used to select a path.

NOTE
Router ID comparison is disabled by default.

To enable router ID comparison, enter the following command at the BGP configuration level of the CLI.

```
BigIron RX(config-bgp)# compare-routerid
```

Syntax: [no] compare-routerid

For more information, refer to "How BGP4 selects a path for a route" on page 740.

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 BGP-related traffic, thus reducing the complexity of the Interior Border Gateway Protocol (IBGP) mesh among the BGP routers in the AS.

The Brocade implementation of this feature is based on RFC 3065.

Normally, all BGP routers within an AS must be fully meshed, so that each BGP router has BGP sessions to all the other BGP routers within the AS. This is feasible in smaller ASs but becomes unmanageable in ASs containing many BGP routers.

When you configure BGP routers into a confederation, all the routers within a sub-AS (a subdivision of the AS) use IBGP and must be fully meshed. However, routers use EBGP to communicate between different sub-ASs.

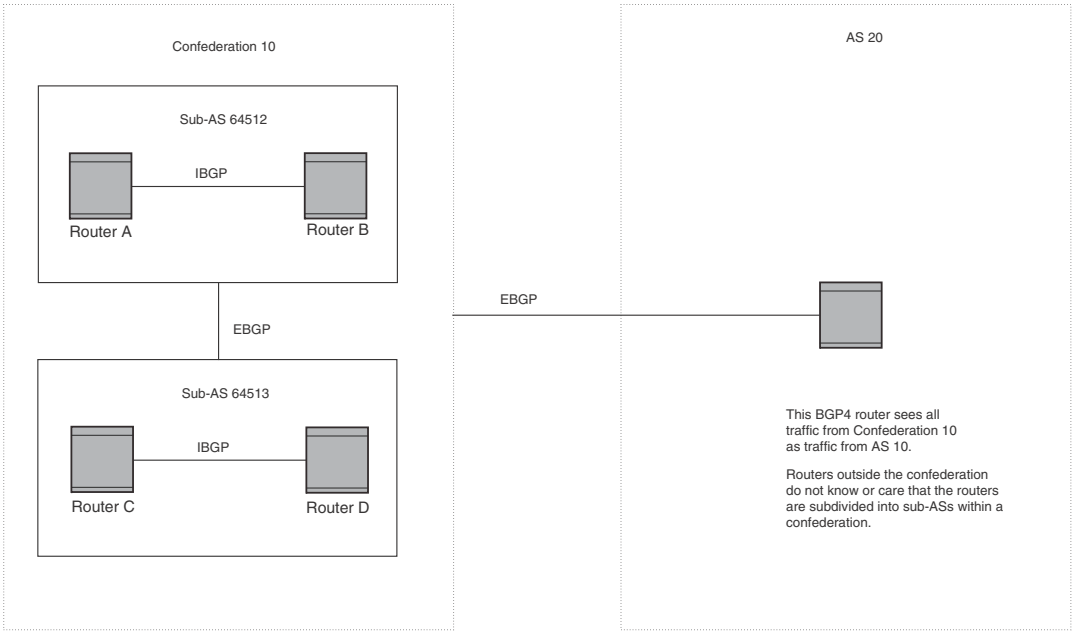
NOTE
Another method for reducing the complexity of an IBGP mesh is to use route reflection. However, if you want to run different Interior Gateway Protocols (IGPs) within an AS, configure a confederation. You can run a separate IGP within each sub-AS.

To configure a confederation, configure groups of BGP routers 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, Brocade recommends that you use numbers from within the private AS range (64512 – 65535). These are private ASs numbers and BGP4 routers do not propagate these AS numbers to the Internet.

Figure 114 shows an example of a BGP4 confederation.

FIGURE 114 Example BGP4 confederation



In this example, four routers are configured into two sub-ASs, each containing two of the routers. The sub-ASs are members of confederation 10. Routers within a sub-AS must be fully meshed and communicate using IBGP. In this example, routers A and B use IBGP to communicate. Routers C and D also use IBGP. However, the sub-ASs communicate with one another using EBGP. For example, router A communicates with router C using EBGP. The routers in the confederation communicate with other ASs using EBGP.

Routers in other ASs are unaware that routers A – D are configured in a confederation. In fact, when routers in confederation 10 send traffic to routers in other ASs, the confederation ID is the same as the AS number for the routers in the confederation. Thus, routers in other ASs see traffic from AS 10 and are unaware that the routers in AS 10 are subdivided into sub-ASs within a confederation.

Configuring a BGP confederation

Perform the following configuration tasks on each BGP router within the confederation:

- Configure the local AS number. The local AS number indicates membership in a sub-AS. All BGP routers with the same local AS number are members of the same sub-AS. BGP routers use the local AS number when communicating with other BGP routers within the confederation.
- Configure the confederation ID. The confederation ID is the AS number by which BGP routers outside the confederation know the confederation. Thus, a BGP router outside the confederation is not aware and does not care that your BGP routers are in multiple sub-ASs. BGP routers use the confederation ID when communicating with routers outside the confederation. The confederation ID must be different from the sub-AS numbers.

- Configure the list of the sub-AS numbers that are members of the confederation. All the routers within the same sub-AS use IBGP to exchange router information. Routers in different sub-ASs within the confederation use EBGP to exchange router information.

The procedures show how to implement the example confederation shown in Figure 26.3.

To configure four BigIron RX devices to be a member of confederation 10, consisting of two sub-ASs (64512 and 64513), enter commands such as the following.

Commands for Router A

```
BigIron RXA(config)# router bgp
BigIron RXA(config-bgp)# local-as 64512
BigIron RXA(config-bgp)# confederation identifier 10
BigIron RXA(config-bgp)# confederation peers 64512 64513
BigIron RXA(config-bgp)# write memory
```

Syntax: local-as <*num*>

The <*num*> parameter with the **local-as** command indicates the AS number for the BGP routers within the sub-AS. You can specify a number from 1 – 65535. Brocade recommends that you use a number within the range of well-known private ASs, 64512 – 65535.

Syntax: confederation identifier <*num*>

The <*num*> parameter with the **confederation identifier** command indicates the confederation number. The confederation ID is the AS number by which BGP routers outside the confederation know the confederation. Thus, a BGP router outside the confederation is not aware and does not care that your BGP routers are in multiple sub-ASs. BGP routers use the confederation ID when communicating with routers outside the confederation. The confederation ID must be different from the sub-AS numbers. You can specify a number from 1 – 65535.

Syntax: 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 may list all sub-ASs in the confederation. Also, you must specify all the sub-ASs with which this router has peer sessions in the confederation. All the routers within the same sub-AS use IBGP to exchange router information. Routers in different sub-ASs within the confederation use EBGP to exchange router information. You can specify a number from 1 – 65535.

Commands for Router B

```
BigIron RXB(config)# router bgp
BigIron RXB(config-bgp)# local-as 64512
BigIron RXB(config-bgp)# confederation identifier 10
BigIron RXB(config-bgp)# confederation peers 64512 64513
BigIron RXB(config-bgp)# write memory
```

Commands for Router C

```
BigIron RXC(config)# router bgp
BigIron RXC(config-bgp)# local-as 64513
BigIron RXC(config-bgp)# confederation identifier 10
BigIron RXC(config-bgp)# confederation peers 64512 64513
BigIron RXC(config-bgp)# write memory
```

Commands for Router D

```
BigIron RXD(config)# router bgp
BigIron RXD(config-bgp)# local-as 64513
BigIron RXD(config-bgp)# confederation identifier 10
BigIron RXD(config-bgp)# confederation peers 64512 64513
BigIron RXD(config-bgp)# write memory
```

Configuring route flap dampening

Route Flap Dampening reduces the amount of change propagated by BGP due to routing state caused by unstable routes. Reducing change propagation will help reduce processing requirements.

To enable route flap dampening using the default values, enter the following command.

```
BigIron RX(config-bgp)# dampening
```

Syntax: dampening [<*half-life*> <*reuse*> <*suppress*> <*max-suppress-time*>]

The <*half-life*> parameter specifies the number of minutes after which the route's penalty becomes half its value. The route penalty allows routes that have remained stable for a while 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. Thus, a dampened route that is no longer unstable can eventually become eligible for use again. You can configure the half-life to be from 1 - 45 minutes. The default is 15 minutes.

The <*reuse*> parameter specifies how low a route's penalty must become 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 a route's penalty can become 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 – 20000 minutes. The default is four times the half-life setting. Thus, if you use the default half-life of 15 minutes, the maximum suppression time is 60 minutes.

The following example shows how to change the dampening parameters.

```
BigIron RX(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. If you want to leave some 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 BigIron RX checks for the existence of an IGP route for 0.0.0.0/0 in the IP route table before creating a local BGP route for 0.0.0.0/0.

To enable the router to originate and advertise a default BGP4 route, enter the following command.

```
BigIron RX(config-bgp)# default-information-originate
```

Syntax: [no] default-information-originate

Changing the default local preference

When the router 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 is an attribute that 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 routers can exchange local preference information with neighbors who also are in the local AS, but BGP4 routers 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 798. Refer to "How BGP4 selects a path for a route" on page 740 for information about the BGP4 algorithm.

To change the default local preference to 200, enter the following command.

```
BigIron RX(config-bgp)# default-local-preference 200
```

Syntax: 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, BGP 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.

```
BigIron RX(config-bgp)# default-metric 40
```

Syntax: default-metric <*value*>

The <*value*> indicates the metric and can be a value from 0 – 4294967295.

Changing administrative distances

BigIron RX can learn about networks from various protocols, including the EBGP portion of BGP4 and IGPs such as OSPF, ISIS, and RIP. Consequently, 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 routers use to compare routes from different sources.

The device re-advertises a learned best BGP4 route to the BigIron RX's 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 BGP selects based on comparison of the paths' BGP4 route parameters. Refer to "How BGP4 selects a path for a route" on page 740.

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 each route's administrative distance. 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.

Here are the default administrative distances on the device:

- **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 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. 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 route's administrative distance is lower than other routes from different route sources to the same destination.

- To change the EBGP, IBGP, and Local BGP default administrative distances, see the instructions in this section.
- To change the default administrative distance for OSPF, RIP, ISIS, refer to "Changing administrative distances" on page 765.
- To change the administrative distance for static routes, refer to "Configuring static routes" on page 191

To change the default administrative distances for EBGP, IBGP, and Local BGP, enter a command such as the following.

```
BigIron RX(config-bgp)# distance 200 200 200
```

Syntax: 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 BGP distance and can be a value from 1 – 255.

Requiring the first AS to be the neighbor's AS

By default, the device does not require the first AS listed in the AS_SEQUENCE field of an AS path Update from an EBGP neighbor to be the AS that the neighbor who sent the Update is in. You can enable the device for this requirement.

When you enable the device to require the AS that an EBGP neighbor is in to 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 ASs match. If the ASs 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.

To enable this feature, enter the following command at the BGP configuration level of the CLI.

```
BigIron RX(config-bgp)# enforce-first-as
```

Syntax: [no] enforce-first-as

Neighbor local-AS

The Neighbor Local Autonomous System (AS) allows a router that is a member of one AS to appear to also be a member of another AS. This feature is useful, for example, if Company A purchases Company B, but Company B does not want to modify it peering configurations.

This feature can only be used for true EBGP peers. When establishing a BGP connection, the router will use the configured neighbor local AS, instead of the system AS number.

For example, if you want a router to use AS 200, instead of 100 when peering with neighbor 11.11.11.2, enter commands such as the following.

```
BigIron RX(config)#router bgp
BigIron RX(config-bgp-router)#local-as 100
BigIron RX(config-bgp-router)#graceful-restart restart-time 30
BigIron RX(config-bgp-router)#graceful-restart
BigIron RX(config-bgp-router)#neighbor 11.11.11.2 remote-as 101
BigIron RX(config-bgp-router)#neighbor 11.11.11.2 local-as 200
```

Syntax: [no] neighbor <*ip-address*> local-as <*local-as-number*>

Enter the IP address of the neighbor with which the device will be peering for <*ip-address*>.

Enabling fast external fallover

BGP4 routers 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 dies, the router will wait until the Hold Time expires or the TCP connection fails before concluding that the neighbor is dead and closing its BGP4 session and TCP connection with the neighbor.

The router waits for the Hold Time to expire before ending the connection to a directly-attached BGP4 neighbor that dies.

For directly attached neighbors, the router immediately senses loss of a connection to the neighbor from a change of state of the port or interface that connects the router to its neighbor. For directly attached EBGP neighbors, the router can use this information to immediately close the BGP4 session and TCP connection to locally attached neighbors that die.

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.

```
BigIron RX(config-bgp)# fast-external-fallover
```

To disable fast external fallover again, enter the following command.

```
BigIron RX(config-bgp)# no fast-external-fallover
```

Syntax: [no] fast-external-fallover

Setting the local AS number

The local AS number identifies the AS the Brocade BGP4 router is in.

To set the local AS number, enter commands such as the following.

```
BigIron RX(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
BigIron RX(config-bgp)# local-as 10
BigIron RX(config-bgp)# write memory
```

Syntax: [no] local-as <*num*>

The <*num*> parameter specifies the local AS number 1 – 65535. There is no default. AS numbers 64512 – 65535 are the well-known private BGP4 AS numbers and are not advertised to the Internet community.

Changing the maximum number of shared BGP4 paths

When IP load sharing is enabled, BGP4 can balance traffic to a specific destination across up to eight equal paths. You can set the maximum number of paths to a value from 1 – 8. The default is 1.

NOTE
The maximum number of BGP4 load sharing paths cannot be greater than the maximum number of IP load sharing paths. To increase the maximum number of IP load sharing paths, use the **ip load sharing** <*num*> command at the global CONFIG level of the CLI or use the # of Paths field next to Load Sharing on the IP configuration panel of the Web management interface.

To change the maximum number of shared paths, enter commands such as the following.

```
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# maximum-paths 4
BigIron RX(config-bgp)# write memory
```

Syntax: [no] maximum-paths <*number*>

The <*num*> parameter specifies the maximum number of paths across which the BigIron RX 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.

Treating missing MEDs as the worst MEDs

By default, the device favors a lower MED over a higher MED during MED comparison. Since the device assigns the value 0 to a route path's MED if the MED value is missing, the default MED comparison results in the device favoring the route paths that are missing their MEDs.

To change this behavior so that the device favors a route that has a MED over a route that is missing its MED, enter the following command at the BGP4 configuration level of the CLI.

```
BigIron RX(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 that is missing its 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.

Customizing BGP4 load sharing

By default, when BGP4 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 BGP configuration level of the CLI.

```
BigIron RX(config-bgp)# multipath ibgp
```

To enable load sharing of EBGP paths only, enter the following command at the BGP configuration level of the CLI.

```
BigIron RX(config-bgp)# multipath ebgp
```

To enable load sharing of paths from different neighboring ASs, enter the following command at the BGP configuration level of the CLI.

```
BigIron RX(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.

Configuring BGP4 neighbors

The BGP4 protocol does not contain a peer discovery process. Therefore, for each of the router's BGP4 neighbors (peers), you must indicate the neighbor's IP address and the AS each neighbor is in. Neighbors that are in different ASs communicate using EBGP. Neighbors within the same AS communicate using IBGP.

NOTE
If the BigIron RX 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 776.

NOTE
The BigIron RX 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 BigIron RX 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 779.

NOTE
When a route-map, prefix-list, or as-path ACL is modified, BGP will be notified. Outbound route polices will be updated automatically. No longer requires user to manually clear neighbor soft-outbound. If the filter is used by BGP inbound route policies, a manual clear of a neighbor is still required.

To add a BGP4 neighbor with IP address 209.157.22.26 remote-as 100, enter the following command.

```
BigIron RX(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 some additional parameters, as shown in the following syntax.

Syntax: [no] neighbor <*ip-addr*> |<*peer-group-name*>
[advertisement-interval <*num*>]
[capability orf prefixlist [send | receive]]
[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]
[timers keep-alive <*num*> hold-time <*num*>]
[unsuppress-map <*map-name*>]
[update-source <*ip-addr*> | ethernet <*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's 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 776.

advertisement-interval <*num*> specifies the minimum delay (in seconds) between messages to the specified neighbor. The default is 30 for EBGP neighbors (neighbors in other ASs). The default is 5 for IBGP neighbors (neighbors in the same AS). The range is 0 – 600.

capability orf prefixlist [**send** | **receive**] configures cooperative router 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 the capability, 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 807.

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** keyword specifies 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 router 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.

Alternatively, you can specify **distribute-list** <*acl-num*> **in | out** to use an IP ACL instead of a distribute list. In this case, <*acl-num*> is an IP ACL.

NOTE
By default, if a route does not match any of the filters, the BigIron RX 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 "Filtering specific IP addresses" on page 749.

ebgp-multihop [<*num*>] specifies that the neighbor is more than one hop away and that the session type with the neighbor is thus 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.

filter-list in | out <*num,num,...*> specifies an AS-path filter list or a list of AS-path ACLs. The in | out keyword specifies 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 router 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 BigIron RX 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 793.

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*> parameter specifies the maximum number. You can specify a value from 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 neighbor's maximum-prefix configuration. The software also generates a Syslog message.

next-hop-self specifies that the router should list itself as the next hop in updates sent to the specified neighbor. This option is disabled by default.

password [0 | 1] <*string*> specifies an MD5 password for securing sessions between the BigIron RX and the neighbor. You can enter a string up to 80 characters long. The string can contain any alphanumeric characters, but the first character cannot be a number. If the password contains a number, do not enter a space following the number.

The **0 | 1** parameter is the encryption option, which you can omit (the default) or which can be one of the following:

- **0** – Disables encryption for the authentication string you specify with the command. The password or string is shown as clear text in the output of commands that display neighbor or peer group configuration information.
- **1** – Assumes that the authentication string you enter is the encrypted form, and decrypts the value before using it.

For more information, refer to "Encryption of BGP4 MD5 authentication keys" on page 774.

NOTE
If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter 0 or 1. Instead, omit the encryption option and allow the software to use the default behavior. If you specify encryption option 1, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option 1 followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

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** keyword specifies 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 797.

remote-as <*as-number*> specifies the AS the remote neighbor is in. The <*as-number*> can be a number from 1 – 65535. There is no default.

remove-private-as configures the router to remove private AS numbers from UPDATE messages the router sends to this neighbor. The router 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** keyword specifies 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 798.

route-reflector-client specifies that this neighbor is a route-reflector client of the router. Use the parameter only if this router is going to be a route reflector. For information, refer to "Configuring a route reflector" on page 759. This option is disabled by default.

send-community enables sending the community attribute in updates to the specified neighbor. By default, the router does not send the community attribute.

shutdown administratively shuts down the session with this neighbor. Shutting down the session allows you to completely configure the neighbor and save the configuration without actually establishing a session with the neighbor. 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's BGP4 route table or resetting the session with the neighbor. Refer to "Using soft reconfiguration" on page 815.

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 from 0 – 65535 seconds. For the Hold Time, you can specify 0 or
3 – 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the router waits indefinitely for messages from a neighbor without concluding that the neighbor is dead. 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 787.

unsuppress-map <*map-name*> removes route suppression from a neighbor's routes when those routes have been suppressed due to aggregation. Refer to "Removing route dampening from suppressed neighbor's routes" on page 773.

update-source <*ip-addr*> | **ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> configures the router 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.

Removing route dampening from suppressed neighbor's routes

You can selectively unsuppress more-specific routes that have been suppressed due to aggregation, and allow the routes to be advertised to a specific neighbor or peer group.

Here is an example.

```
BigIron RX(config-bgp)# aggregate-address 209.1.0.0 255.255.0.0 summary-only
BigIron RX(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 the example above, 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 the command below, the display shows that the route is not being advertised to the BigIron RX's BGP4 neighbors.

```
BigIron RX(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
```

If you want to override the **summary-only** parameter and allow a specific route to be advertised to a neighbor, enter commands such as the following.

```
BigIron RX(config)# ip prefix-list Unsuppress1 permit 209.1.44.0/24
BigIron RX(config)# route-map RouteMap1 permit 1
BigIron RX(config-routemap RouteMap1)# match prefix-list Unsuppress1
BigIron RX(config-routemap RouteMap1)# exit
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# neighbor 10.1.0.2 unsuppress-map RouteMap1
BigIron RX(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 following command verifies that the route has been unsuppressed.

```
BigIron RX(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)
```

Encryption of BGP4 MD5 authentication keys

When you configure a BGP4 neighbor or neighbor peer group, you can specify an MD5 authentication string for authenticating packets exchanged with the neighbor or peer group of neighbors.

For added security, the software encrypts display of the authentication string by default. The software also provides an optional parameter to disable encryption of the authentication string, on an individual neighbor or peer group basis. By default, the 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.

In addition, when you save the configuration to the startup configuration file, the file contains the new BGP4 command syntax and encrypted passwords or strings.

NOTE
Brocade recommends that you save a copy of the startup configuration file for each BigIron RX you plan to upgrade.

Encryption example

The following commands configure a BGP4 neighbor and a peer group, and specify MD5 authentication strings (passwords) for authenticating packets exchanged with the neighbor or peer group.

```
BigIron RX(config-bgp)# local-as 2
BigIron RX(config-bgp)# neighbor xyz peer-group
BigIron RX(config-bgp)# neighbor xyz password abc
BigIron RX(config-bgp)# neighbor 10.10.200.102 peer-group xyz
BigIron RX(config-bgp)# neighbor 10.10.200.102 password test
```

Here is how the commands appear when you display the BGP4 configuration commands.

```
BigIron RX(config-bgp)# show ip bgp config
Current BGP configuration:
router bgp
 local-as 2
 neighbor xyz peer-group
 neighbor xyz password 1 $!2d
 neighbor 10.10.200.102 peer-group xyz
 neighbor 10.10.200.102 remote-as 1
 neighbor 10.10.200.102 password 1 $on-o
```

Notice that 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 [0 | 1] <*string*>

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.

The password <*string*> parameter specifies an MD5 authentication string for securing sessions between the device and the neighbor. You can enter a string up to 80 characters long. The string can contain any alphanumeric characters, but the first character cannot be a number. If the password contains a number, do not enter a space following the number.

The **0 | 1** parameter is the encryption option, which you can omit (the default) or which can be one of the following:

- **0** – Disables encryption for the authentication string you specify with the command. The password or string is shown as clear text in the output of commands that display neighbor or peer group configuration information.

- **1** – Assumes that the authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE
If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter 0 or 1. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option 1, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option 1 followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

Displaying the authentication string

If you want to display the authentication string, enter the following commands:

```
BigIron RX(config)# enable password-display
BigIron RX(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. 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.

NOTE
The command also displays SNMP community strings in clear text, in the output of the **show snmp server** command.

Configuring a BGP4 peer group

A ***peer group*** is a set of BGP4 neighbors that share common parameters. Peer groups provide the following benefits:

- **Simplified neighbor configuration** – You can configure a set of neighbor parameters and then apply them to multiple neighbors. You do not need to individually 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 BGP 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 checks to ensure that the peer group does not contain any neighbors. If the peer group does contain neighbors, the software does not allow you to remove the remote AS. The software prevents removing the remote AS in this case so that the neighbors in the peer group that are using the remote AS do not lose connectivity to the BigIron RX.

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 BGP configuration level.

```
BigIron RX(config-bgp)# neighbor PeerGroup1 peer-group
BigIron RX(config-bgp)# neighbor PeerGroup1 description “EastCoast Neighbors”
BigIron RX(config-bgp)# neighbor PeerGroup1 remote-as 100
BigIron RX(config-bgp)# neighbor PeerGroup1 distribute-list out 1
```

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 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*>
[advertisement-interval <*num*>]
[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]
[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 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's 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 769 and "Configuring a BGP4 peer group" on page 776.

The remaining parameters are the same ones supported for individual neighbors. Refer to "Configuring BGP4 neighbors" on page 769 and "Configuring a BGP4 peer group" on page 776.

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.

```
BigIron RX(config-bgp)# neighbor 192.168.1.12 peer-group PeerGroup1
BigIron RX(config-bgp)# neighbor 192.168.2.45 peer-group PeerGroup1
BigIron RX(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: 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 new option to shut down a neighbor, the option takes place immediately and remains in effect until you remove the option. If you save the configuration to the startup configuration file, the shutdown option remains in effect even after a software reload.

NOTE
The software also contains an option to end the session with a BGP4 neighbor and thus 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 thus can prevent the BigIron RX 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 BigIron RX, check the BigIron RX’s running configuration and startup configuration files 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.

```
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# neighbor 209.157.22.26 shutdown
BigIron RX(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 router sends BGP4 routes only for the networks you either identify with the **network** command or are redistributed into BGP from OSPF, ISIS, RIP, or connected routes.

NOTE
The exact route must exist in the IP route table before the BigIron RX can create a local BGP route.

To configure the device to advertise network 209.157.22.0/24, enter the following command.

```
BigIron RX(config-bgp)# network 209.157.22.0 255.255.255.0
```

Syntax: 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; otherwise, 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 BGP weight (200 by default), thus tagging the route as a backdoor route. Use this parameter when you want the router to prefer IGP routes such as RIP or OSPF routes over the EBGP route for the network.

Specifying a route map name when configuring BGP4 network information

You can specify a route map as one of the parameters when you configure a BGP4 network to be advertised. The device can use 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 BGP.

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.

```
BigIron RX(config)# route-map set_net permit 1
BigIron RX(config-routemap set_net)# set community no-export
BigIron RX(config-routemap set_net)# exit
BigIron RX(config)# router bgp
BigIron RX(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: 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 798.

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 router, you might want to 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.

```
BigIron RX(config-bgp)# next-hop-enable-default
```

Syntax: [no] next-hop-enable-default

Enabling next-hop recursion

For each BGP4 route a BigIron RX learns, the device performs a route lookup to obtain the IP address of the route's next hop. A BGP4 route becomes eligible for installation into 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 Interior Gateway Protocol (IGP) path or a static route path.

By default, the software performs only one lookup for a BGP route's next-hop IP address. 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 BGP path, the software considers the BGP route's destination to be unreachable. The route is not eligible to be installed in the IP route table.

It is possible for the BGP route table to contain a route whose next-hop IP address is not reachable through an IGP route, even though a hop farther away can be reached by the device through an IGP route. This can occur when the IGPs do not learn a complete set of IGP routes, resulting in the device learning about an internal route through IBGP instead of through an IGP. In this case, the IP route table does not contain a route that can be used to reach the BGP route's destination.

To enable the device to find the IGP route to a BGP route's next-hop gateway, enable recursive next-hop lookups. When you enable recursive next-hop lookup, if the first lookup for a BGP route results in an IBGP path originated within the same Autonomous System (AS), rather than an IGP path or static route path, the device performs a lookup on the next-hop gateway's next-hop IP address. If this second lookup results in an IGP path, the software considers the BGP route to be valid and thus eligible for installation in the IP route table. Otherwise, the device performs a lookup on the next-hop IP address of the next-hop gateway's next hop, 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

Here is an example of the results of an unsuccessful next-hop lookup for a BGP route. In this case, next-hop recursive lookups are disabled. The example is for the BGP route to network 240.0.0.0/24.

```
BigIron RX# 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 thus is considered unreachable by the device. Here is the IP route table entry for the BGP route's next-hop gateway (102.0.0.1/24).

```
BigIron RX# 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
```

The route to the next-hop gateway is a BGP route, not an IGP route, and thus 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 BGP route's next-hop gateway.

```
BigIron RX# 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 recursively looks up the next-hop gateways along the route until the device finds an IGP route to the BGP route's destination. Here is an example.

```
BigIron RX# 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.

```
BigIron RX# 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 thus cannot use the BGP route. In this case, since recursive next-hop lookups are enabled, the device next performs a lookup for 102.0.0.1's next-hop gateway, 10.0.0.1.

```
BigIron RX# 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 BGP route's destination still cannot be reached through IP. The recursive next-hop lookup feature performs a lookup on 10.0.0.1's next-hop gateway:

```
BigIron RX# 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. In fact, this route is a directly-connected route. As a result, the BGP route's destination is now reachable through IGP, which means the BGP route is eligible for installation in the IP route table. Here is the BGP route in the IP route table.

```
BigIron RX# 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
```

This device can use this route because the device 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 BGP configuration level of the CLI.

```
BigIron RX(config-bgp)# next-hop-recursion
```

Syntax: [no] next-hop-recursion

Modifying redistribution parameters

By default, the router does not redistribute route information between BGP4 and the IP IGPs (RIP, ISIS, and OSPF). You can configure the router 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.

```
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# redistribute ospf
BigIron RX(config-bgp)# redistribute connected
BigIron RX(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 BGP.

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

NOTE
Entering **redistribute ospf** simply redistributes internal OSPF routes. If you want to redistribute external OSPF routes also, you must use the **redistribute ospf match external...** command. Refer to "Redistributing OSPF external routes" on page 785.

NOTE
When a route-map, prefix-list, or as-path ACL is modified, BGP will be notified. Outbound route polices will be updated automatically. No longer requires user to manually clear neighbor soft-outbound. If the filter is used by BGP inbound route policies, a manual clear of a neighbor is still required.

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 BGP.

Redistributing connected routes

To configure BGP4 to redistribute directly connected routes, enter the following command.

```
BigIron RX(config-bgp)# redistribute connected
```

Syntax: 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 router. Refer to "Defining route maps" on page 798 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.

```
BigIron RX(config-bgp)# redistribute rip metric 10
```

Syntax: 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 router. Refer to "Defining route maps" on page 798 for information about defining route maps.

Redistributing OSPF external routes

To configure the BigIron RX to redistribute OSPF external type 1 routes, enter the following command.

```
BigIron RX(config-bgp)# redistribute ospf match external1
```

Syntax: 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** parameter applies only to OSPF. This parameter specifies 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 **redistribute 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 router. Refer to "Defining route maps" on page 798 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 ISIS

To configure the device to redistribute ISIS routes, enter the following command.

```
BigIron RX(config-bgp)# redistribute isis level-1
```

Syntax: redistribute isis level-1 | level-1-2 | level-2 [metric <*num*>] [route-map <*map-name*>]

The **isis** parameter indicates that you are redistributing ISIS routes into BGP4.

The **level-1** parameter redistributes ISIS routes only within the area the routes.

The **level-2** parameter redistributes ISIS routes between areas within a domain.

The **level-1-2** parameter redistributes ISIS 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.

```
BigIron RX(config-bgp)# redistribute static
```

Syntax: 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** <*map-name*> parameter specifies a route map to be consulted before adding the static route to the BGP4 route table.

The route map you specify must already be configured on the router. Refer to "Defining route maps" on page 798 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), this means that 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 configure the route map, then identify it as a table map. The table map does not require separate configuration. You create it simply by calling an existing route map a table map. 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. This route map is then identified as a table map. As a result, the route map is applied only to routes that 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.

```
BigIron RX(config)# route-map TAG_IP permit 1
BigIron RX(config-routemap TAG_IP)# match ip address prefix-list p11
BigIron RX(config-routemap TAG_IP)# set tag 100
BigIron RX(config-routemap TAG_IP)# router bgp
BigIron RX(config-bgp)# table-map TAG_IP
```

Changing the keep alive time and hold time

The Keep Alive Time specifies how frequently the router will send KEEPALIVE messages to its BGP4 neighbors. The Hold Time specifies how long the router will wait for a KEEPALIVE or UPDATE message from a neighbor before concluding that the neighbor is dead. When the router concludes that a BGP4 neighbor is dead, the router 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 769 and "Configuring a BGP4 peer group" on page 776.

To change the Keep Alive Time to 30 and Hold Time to 90, enter the following command.

```
BigIron RX(config-bgp)# timers keep-alive 30 hold-time 90
```

Syntax: 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 router 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 its 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, enter a command such as the following at the BGP configuration level of the CLI.

```
BigIron RX(config-bgp)# update-time 15
```

This command changes the update timer to 15 seconds.

Syntax: [no] update-time <*secs*>

The <*secs*> parameter specifies the number of seconds and can be from 1 – 30. The default is 5.

Changing the router ID

The OSPF and BGP4 protocols use router IDs to identify the routers that are running the protocols. A router ID is a valid, unique IP address and sometimes is an IP address configured on the router. The router ID cannot be an IP address in use by another device.

By default, the router ID on a BigIron RX 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 device. 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 address configured on the device.

NOTE
A BigIron RX 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** CLI command at any CLI level.

To change the router ID, enter a command such as the following.

```
BigIron RX(config)# ip router-id 209.157.22.26
```

Syntax: 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 BigIron RX, but do not specify an IP address in use by another device.

Adding a loopback interface

You can configure the router 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 router and its 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 router. When you configure a BGP4 neighbor on the router, you can specify whether the router uses the loopback interface to communicate with the neighbor. As long as a path exists between the router and its neighbor, BGP4 information can be exchanged. The BGP4 session is not associated with a specific link but instead is associated with the virtual interfaces.

NOTE
If you configure the BigIron RX to use a loopback interface to communicate with a BGP4 neighbor, the peer IP address on the remote router pointing to your loopback address must be configured.

To add a loopback interface, enter commands such as the following.

```
BigIron RX(config-bgp)# exit
BigIron RX(config)# int loopback 1
BigIron RX(config-lbif-1)# ip address 10.0.0.1/24
```

Syntax: interface loopback <*num*>

The <*num*> value can be from 1 – 8.

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 type (EBGP or IBGP) for the route.

To configure the device to perform BGP4 load sharing:

- Enable IP load sharing if it is disabled.
- Set the maximum number of paths. The default maximum number of BGP4 load sharing paths is 1, which means no BGP4 load sharing takes place by default. Refer to "Changing the maximum number of shared BGP4 paths" on page 768.

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 last comparison the device performs is a comparison of the internal paths.

- When IP load sharing is disabled, the device prefers the path to the router with the lower router ID if the **compare-routerid** command is enabled.
- When IP load sharing and BGP4 load sharing are enabled, the device balances the traffic across the multiple paths instead of choosing just one path based on router ID.

Refer to "How BGP4 selects a path for a route" on page 740 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 of IP load sharing paths to a value from 2 – 8.

For more information on how load sharing works on the device, refer to "Configuring IP load sharing" on page 201.

Configuring route reflection parameters

Normally, all the BGP routers within an AS are fully meshed. Each of the routers has an IBGP session with each of the other BGP routers in the AS. Each IBGP router thus has a route for each of its IBGP neighbors. For large ASs containing many IBGP routers, the IBGP route information in each of the fully-meshed IBGP routers can introduce too much administrative overhead.

To avoid this problem, you can hierarchically organize your IGP routers into clusters.

- A ***cluster*** is a group of IGP routers 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 the configuration for route reflection takes place on the route reflectors. The 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 router 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 router configured to send BGP route information to all the clients (other BGP4 routers) within the cluster. Route reflection is enabled on all Brocade BGP4 routers by default but does not take effect unless you add route reflector clients to the router.

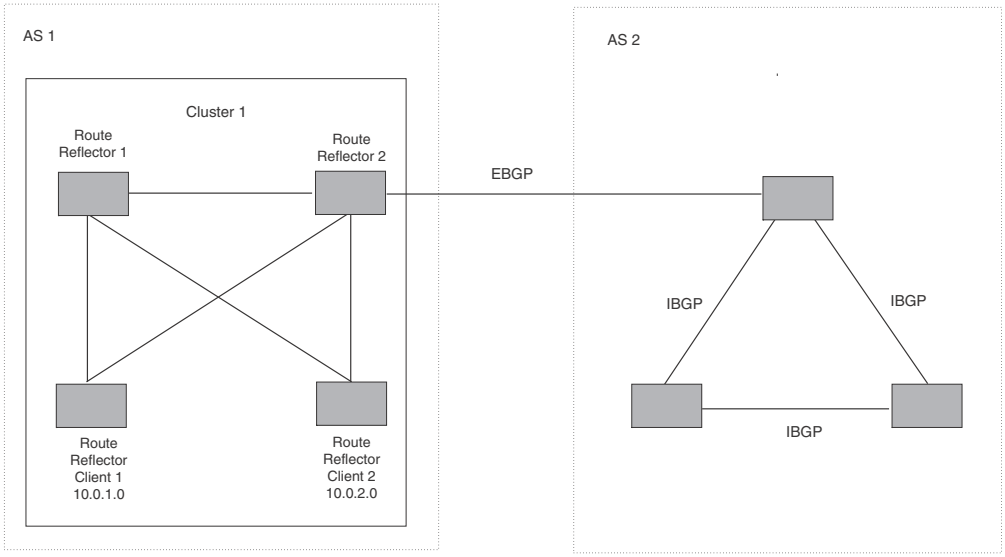
- A ***route reflector client*** is an IGP router identified as a member of a cluster. You identify a router as a route reflector client on the router that is the route reflector, not on the client. The client itself requires no additional configuration. In fact, the client does not know that it is a 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 routers 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 BigIron RX devices are configured as route reflectors for the same cluster. The route reflectors provide redundancy in case one of the reflectors becomes unavailable. Without redundancy, if a route reflector becomes unavailable, its clients 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 routers, but the clients are not fully meshed. They rely on the route reflectors to propagate BGP4 route updates.

FIGURE 115 Example route reflector configuration



Support for RFC 2796

Route reflection is based on RFC 2796. This updated RFC helps eliminate routing loops that are possible in some implementations of the older specification, RFC 1966.

- 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 router ID of the router that originated the route. Moreover, the route reflector sets the attribute only if this is the first time the route is being reflected (sent by a route reflector).

- If a device receives a route whose ORIGINATOR_ID attribute has the value of the BigIron RX's own router ID, the BigIron RX 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 who receive the route from an IBGP neighbor add their cluster IDs to the front of the route's CLUSTER_LIST. If the route reflector does not have a cluster ID configured, the device adds its router ID to the front of the CLUSTER_LIST.
- If BigIron RX configured as a route reflector receives a route whose CLUSTER_LIST contains the route reflector's own cluster ID, the route reflector discards the route and does not forward it.

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 BigIron RX as route reflector 1 in Figure 26.4 on page 26-42. 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.

```
BigIron RX(config-bgp)# cluster-id 1
BigIron RX(config-bgp)# neighbor 10.0.1.0 route-reflector-client
BigIron RX(config-bgp)# neighbor 10.0.2.0 route-reflector-client
```

Syntax: [no] cluster-id <*num*> | <*ip-addr*>

The <*num*> | <*ip-addr*> parameter specifies the cluster ID and can be a number from 1 – 4294967295 or an IP address. The default is the router ID. You can configure one cluster ID on the router. All route-reflector clients for the router 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 within the cluster.

To add an IBGP neighbor to the cluster, enter the following command.

Syntax: neighbor <*ip-addr*> route-reflector-client

For more information about the **neighbor** command, refer to "Configuring BGP4 neighbors" on page 769 and "Configuring a BGP4 peer group" on page 776.

Filtering

This section describes the following:

- "Using a table map to set the tag value" on page 787
- "Configuring cooperative BGP4 route filtering" on page 807

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, if you want 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 to deny such routes.

The device provides the following methods for filtering on AS-path information:

- AS-path filters - refer to "Defining an AS-path filter" on page 751.
- AS-path ACLs

NOTE
The BigIron RX cannot actively 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 a BGP neighbor's filter list number as well as by match statements 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.

```
BigIron RX(config)# ip as-path access-list acl1 permit 100
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# neighbor 10.10.10.1 filter-list 1 in
```

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.

Syntax: ip as-path access-list <*string*> [seq <*seq-value*>] deny | permit <*regular-expression*>

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 AS-path list's sequence number. If you do not specify a sequence number, the software numbers them 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 a route's AS-path list matches a match statement in this ACL. To configure the AS-path match statements in a route map, use the **match as-path** command. Refer to "Matching based on AS-path ACL" on page 802.

The <*regular-expression*> parameter specifies the AS path information you want to permit or deny to routes that match any of the match statements 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 769 and "Configuring a BGP4 peer group" on page 776.

Using regular expressions

You 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.

In addition, you can 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.

```
BigIron RX(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.

```
BigIron RX(config-bgp)# ip as-path access-list acl1 permit [xyz]
```

Special characters

When you enter as single-character expression or a list of characters, you also can use the following special characters. Table 26.2 on page 26-45 lists the special characters. The description for each special character includes an example. Notice that you place some special characters in front of the characters they control but you place other special characters after the characters they control. In each case, the examples show where to place the special character.

TABLE 117 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
$	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$

TABLE 117 BGP4 special characters for regular expressions (Continued)

Character	Operation
_	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. See 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. See 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)?)

If you want to filter for a special character instead of using the special character as described in Table 26.2 on page 26-45, 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 “*”.

```
BigIron RX(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 “\\”.

```
BigIron RX(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 one of a route's attributes. 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 "Defining a community filter" on page 751.
- Community list ACLs

NOTE
The BigIron RX 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 statements in a route map.

Defining a community ACL

To configure community ACL 1, enter a command such as the following.

```
BigIron RX(config)# ip community-list 1 permit 123:2
```

This command configures a community ACL that permits routes that contain community 123:2.

NOTE
Refer to "Matching based on community ACL" on page 802 for information about how to use a community list as a match condition in a route map.

Syntax: ip community-list standard <*string*> [seq <*seq-value*>] deny | permit <*community-num*>

Syntax: 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 community ACL or an extended one. A standard community ACL does not support regular expressions whereas an extended one does. This is the only difference between standard and extended IP community lists.

The **seq** <*seq-value*> parameter is optional and specifies the community list's sequence number. You can configure up to 199 entries in a community list. If you do not specify a sequence number, the software numbers them 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** parameter specifies the action the software takes if a route's community list matches a match statement in this ACL. To configure the community-list match statements in a route map, use the **match community** command. Refer to "Matching based on community ACL" on page 802

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 routers 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 794. 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.

```
BigIron RX(config)# ip prefix-list Routesfor20 permit 20.20.0.0/24
BigIron RX(config)# router bgp
BigIron RX(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: 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. You 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 IP 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 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's route is in this prefix list.

The prefix-list matches only on this network unless you use the **ge** <*ge-value*> or **le** <*le-value*> parameters. (See below.)

The <*network-addr*>/<*mask-bits*> parameter specifies 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*>.

- If you specify only **ge** <*ge-value*>, then the mask-length range is from <*ge-value*> to 32.
- If you specify only **le** <*le-value*>, then 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 specify with the <*network-addr*>/<*mask-bits*> parameter.

For the syntax of the **neighbor** command shown in the example above, refer to "Configuring BGP4 neighbors" on page 769 and "Configuring a BGP4 peer group" on page 776.

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.

```
BigIron RX(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: 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** parameter specifies 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 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 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.

Match statements compare the route against one or more of the following:

- The route's 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 router
- The route's tag
- For OSPF routes only, the route's 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's set statements can perform one or more of the following modifications to the route's attributes:

- Prepend AS numbers to the front of the route's AS-path. By adding AS numbers to the AS-path, you can cause the route to be less preferred when compared to other routes on the basis of the length of the AS-path.
- Add a user-defined tag to the route or add 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 router.
- Set the origin to IGP or INCOMPLETE.
- Set the weight.

For example, when you configure parameters for redistributing routes into BGP, one of the optional parameters is a route map. If you specify a route map as one of the redistribution parameters, the router will match the route against the match statements in the route map. If a match is found and if the route map contains set statements, the router will set attributes in the route according to the set statements.

To create a route map, you define instances of the map. Each instance is identified 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.

```
BigIron RX(config)# route-map GET_ONE permit 1
BigIron RX(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 statements at this level. Refer to "Specifying the match conditions" on page 800 and "Setting parameters in the routes" on page 804.

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 router 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 statements 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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config-routemap GET_ONE)# match address-filters 11
```

Syntax: match
[as-path <*name*>] |
[address-filters | as-path-filters | community-filters <*num,num,...*>] |
[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*>]

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 793.

The **address-filters | as-path-filters | community-filters** <*num,num,...*> parameter specifies a filter or list of filters to be matched for each route. The router treats the first match as the best match. If a route does not match any filter in the list, then the router considers the match condition to have failed. To configure these types of filters, use commands at the BGP configuration level:

- To configure an address filter, refer to "Filtering specific IP addresses" on page 749.
- To configure an AS-path filter or AS-path ACL, refer to "Filtering AS-paths" on page 793.
- To configure a community filter or community ACL, refer to "Filtering communities" on page 795.

You can enter up to six community names on the same command line.

NOTE
The filters must already be configured.

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's community attributes field contains the same community numbers specified in the match statement.

The **ip address | next-hop** <*acl-num*> | prefix-list <*string*> parameter specifies 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 Chapter 21, "Access Control List". To configure an IP prefix list, use the **ip prefix-list** command. Refer to "Defining and applying IP prefix lists" on page 797.

The **ip route-source** <*acl*> | **prefix** <*name*> parameter matches 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's MED (metric) to the specified value.

The **next-hop** <*address-filter-list*> parameter compares the IP address of the route's next hop to the specified IP address filters. The filters must already be configured.

The **route-type internal | external-type1 | external-type2** parameter applies only to OSPF routes. This parameter compares the route's type to the specified value. The **level-1** parameter compares ISIS routes only with routes within the same area. The **level-2** parameter compares ISIS routes only with routes in different areas, but within a domain. The **level-1-2** parameter compares ISIS routes with routes the same area and in different areas, but within a domain.

The **tag** <*tag-value*> parameter compares the route's tag to the specified tag value.

The following sections are some 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.

```
BigIron RX(config)# route-map PathMap permit 1
BigIron RX(config-routemap PathMap)# match as-path 1
```

Syntax: 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 793.

Matching based on community ACL

To construct a route map that matches based on community ACL 1, enter the following commands.

```
BigIron RX(config)# ip community-list 1 permit 123:2
BigIron RX(config)# route-map CommMap permit 1
BigIron RX(config-routemap CommMap)# match community 1
```

Syntax: 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 796.

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.

```
BigIron RX(config)# route-map NetMap permit 1
BigIron RX(config-routemap NetMap)# match ip address 1
```

Syntax: match ip address <*ACL-name-or-num*>

Syntax: 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 Chapter 21, "Access Control List".

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 797.

Matching based on next-hop router

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 router, enter commands such as the following.

```
BigIron RX(config)# route-map HopMap permit 1
BigIron RX(config-routemap HopMap)# match ip next-hop 2
```

Syntax: match ip next-hop <*num*>

Syntax: 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 Chapter 21, "Access Control List".

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 797.

Matching based on the route source

To match a BGP4 route based on its source, use the **match ip route-source** statement. Here is an example.

```
BigIron RX(config)# access-list 10 permit 192.168.6.0 0.0.0.255
BigIron RX(config)# route-map bgp1 permit 1
BigIron RX(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 statement 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: match ip route-source <*acl*> | prefix <*name*>

The <*acl*> | prefix <*name*> parameter specifies the name or ID of an IP ACL, or an IP prefix list.

Matching on routes containing a specific set of communities

BigIron RX enables you to 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.

Here is an example.

```
BigIron RX(config)# ip community-list standard std_1 permit 12:34 no-export
BigIron RX(config)# route-map bgp2 permit 1
BigIron RX(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: 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.

Here is another example.

```
BigIron RX(config)# ip community-list standard std_2 permit 23:45 56:78
BigIron RX(config)# route-map bgp3 permit 1
BigIron RX(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's communities must be the same as those in exactly one of the community ACLs used by the match community statement.

Setting parameters in the routes

Use the following command to define a set statement that prepends an AS number to the AS path on each route that matches the corresponding match statement.

```
BigIron RX(config-routemap GET_ONE)# set as-path prepend 65535
```

Syntax: 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 <*tag-value*>] |
[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 **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 a BGP4 route's community attributes field.

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's penalty becomes half its value. The <*reuse*> parameter specifies how low a route's penalty must become before the route becomes eligible for use again after being suppressed. The <*suppress*> parameter specifies how high a route's 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 763.

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 route's metric to the number you specify.
- **set metric +**<*num*> – Increases route's metric by the number you specify.
- **set metric -**<*num*> – Decreases route's 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's 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's next hop router.

The **origin igp | incomplete** parameter sets the route's origin to IGP or INCOMPLETE.

The **tag** <*tag-value*> parameter sets the route's tag. You can specify a tag value from 0 – 4294967295.

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 BigIron RX places the route in the IP route table instead of changing the value in the BGP route table. Refer to "Using a table map to set the tag value" on page 787.

The **weight** <*num*> parameter sets the weight for the route. You can specify a weight value from 0 – 4294967295.

Setting a BP4 route's MED to be equal to 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.

```
BigIron RX(config)# access-list 1 permit 192.168.9.0 0.0.0.255
BigIron RX(config)# route-map bgp4 permit 1
BigIron RX(config-routemap bgp4)# match ip address 1
BigIron RX(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: 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.

```
BigIron RX(config)# route-map bgp5 permit 1
BigIron RX(config-routemap bgp5)# match ip address 1
BigIron RX(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: 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's 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.

```
BigIron RX(config)# ip community-list standard std_3 permit 12:99 12:86
BigIron RX(config)# route-map bgp6 permit 1
BigIron RX(config-routemap bgp6)# match ip address 1
BigIron RX(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: set comm-list <*acl*> delete

The <*acl*> parameter specifies the name of a community list ACL.

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 its 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 its BGP4 neighbor:

- Configure the filter.

  NOTE
  The current release supports cooperative filtering only for filters configured using IP prefix lists.

- Apply the filter as in ***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 BigIron RX 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.

```
BigIron RX(config)# ip prefix-list Routesfrom1234 deny 20.20.0.0/24
BigIron RX(config)# ip prefix-list Routesfrom1234 permit 0.0.0.0/0 le 32
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# neighbor 1.2.3.4 prefix-list Routesfrom1234 in
BigIron RX(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 also is configured for cooperative filtering.)

Syntax: [no] neighbor <*ip-addr*> | <*peer-group-name*> capability orf prefixlist [send | receive]

The <*ip-addr*> | <*peer-group-name*> parameter specifies the IP address of a neighbor or the name of a peer group of neighbors.

The **send** | **receive** parameter specifies 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 BigIron RX 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.

```
BigIron RX# 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.

```
BigIron RX# 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 BigIron RX 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.

```
BigIron RX# 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.

```
BigIron RX# 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 the change in a route’s state, from up to down or down to up. When a route’s state changes, the state change causes changes in the route tables of the routers that support the route. Frequent changes in a route’s state can cause Internet instability and add processing overhead to the routers that support the route.

Route flap dampening is a mechanism that reduces the impact of route flap by changing a BGP4 router’s response to route state changes. When route flap dampening is configured, the device suppresses unstable routes until the route’s state changes reduce enough to meet an acceptable degree of stability. The Brocade 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 BigIron RX 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 also stops advertising it to other routers. The mechanism also allows a route's penalties to reduce over time if the route's 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 has a penalty value greater than 2000, the device stops using the route. Thus, 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 (it does not go down again) 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's 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.

To configure address filters and a route map for dampening specific routes, enter commands such as the following.

```
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# address-filter 9 permit 209.157.22.0 255.255.255.0
255.255.255.0 255.255.255.0
BigIron RX(config-bgp)# address-filter 10 permit 209.157.23.0 255.255.255.0
255.255.255.0 255.255.255.0
BigIron RX(config-bgp)# exit
BigIron RX(config)# route-map DAMPENING_MAP permit 9
BigIron RX(config-routemap DAMPENING_MAP)# match address-filters 9
BigIron RX(config-routemap DAMPENING_MAP)# set dampening 10 200 2500 40
BigIron RX(config-routemap DAMPENING_MAP)# exit
BigIron RX(config)# route-map DAMPENING_MAP permit 10
BigIron RX(config-routemap DAMPENING_MAP)# match address-filters 10
BigIron RX(config-routemap DAMPENING_MAP)# set dampening 20 200 2500 60
BigIron RX(config-routemap DAMPENING_MAP)# router bgp
BigIron RX(config-bgp)# dampening route-map DAMPENING_MAP
```

The **address-filter** commands in this example configure two BGP4 address filters, for networks 209.157.22.0 and 209.157.23.0. The first route-map command creates an entry in a route map called "DAMPENING_MAP". Within this entry of the route map, the match command matches based on address filter 9, and the **set** command sets the dampening parameters for the route that matches. Thus, for BGP4 routes to 209.157.22.0, the device uses the route map to set the dampening parameters. These parameters override the globally configured dampening parameters.

The commands for the second entry in the route map (instance 10 in this example) perform the same functions for route 209.157.23.0. Notice that the dampening parameters are different for each route.

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 statements. 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 BGP configuration level.
- Configure another route map that explicitly enables dampening. Use a set statement 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 statements 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.

```
BigIron RX(config)# route-map DAMPENING_MAP_ENABLE permit 1
BigIron RX(config-routemap DAMPENING_MAP_ENABLE)# exit
BigIron RX(config)# route-map DAMPENING_MAP_NEIGHBOR_A permit 1
BigIron RX(config-routemap DAMPENING_MAP_NEIGHBOR_A)# set dampening
```

```
BigIron RX(config-routemap DAMPENING_MAP_NEIGHBOR_A)# exit
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# dampening route-map DAMPENING_MAP_ENABLE
BigIron RX(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 statements. At the BGP 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 statement. The route map implicitly applies to all routes. Since the route map will be applied to a neighbor at the BGP 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 statements for specific routes, the route map enables dampening for all routes received from the neighbor.

Removing route dampening from a route

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.

```
BigIron RX# 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.

```
BigIron RX# 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 level of the CLI.

```
BigIron RX# 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. The regular expressions are the same ones supported for BGP4 AS-path filters. Refer to "Using regular expressions" on page 794.

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 **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: <*ip-addr*> flap-statistics.

This display shows the following information.

TABLE 118 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the BigIron RX'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 neighbor that sent the route to the BigIron RX.
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	Shows the AS-path information for the route.

You also can display all the dampened routes by entering the following command.
show ip bgp dampened-paths.

Clearing route flap dampening statistics

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 any level of the CLI.

```
BigIron RX# 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 763.

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 763.

Generating traps for BGP

BigIron RX provides the ability to enable and disable SNMP traps for BGP. BGP traps are enabled by default.

To enable BGP traps after they have been disabled, enter the following command.

```
BigIron RX(config)# snmp-server enable traps bgp
```

Syntax: [no] snmp-server enable traps bgp

Use the **no** form of the command to disable BGP traps.

Updating route information and resetting a neighbor session

The following sections describe ways to update route information with a neighbor, reset the 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, for existing outbound routes, you must reset the neighbor to update the outbound routes.

Similar to inbound 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 routers today support this capability.
- Clear (reset) the session with the neighbor or peer group. This is the only method you can use if the 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 820.

Using soft reconfiguration

The ***soft reconfiguration*** feature places policy changes into effect without resetting the BGP4 session. Soft reconfiguration does not request the neighbor or group to send its entire BGP4 table, nor does the feature reset the session with the neighbor or group. Instead, the soft reconfiguration feature 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's 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.

```
BigIron RX(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 refreshes or 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 769 and "Configuring a BGP4 peer group" on page 776.

Placing a policy change into effect

To place policy changes into effect, enter a command such as the following.

```
BigIron RX(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 818.

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. This includes 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.

```
BigIron RX# 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 by the command are the routes that the BigIron RX's BGP4 policies filtered out. 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 above 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 the 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 839.

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.

```
BigIron RX# 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. The example above shows summary information.

NOTE
The syntax for displaying received routes is shown. For complete command syntax, refer to "Displaying BGP4 neighbor information" on page 827.

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 router 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 dynamically refresh BGP4 routes to place new or changed filters into effect.

To request a dynamic refresh of all routes from a neighbor, enter a command such as the following.

```
BigIron RX(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*> specifies 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 815.
  - If you did not enable soft reconfiguration, **soft in** requests the neighbor's entire BGP4 route table (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 device's entire BGP4 route table (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 BigIron RX's entire BGP4 route table (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 BigIron RX's BGP4 routes to a neighbor, enter a command such as the following.

```
BigIron RX(config-bgp)# clear ip bgp neighbor 192.168.1.170 soft out
```

This command applies its filters for outgoing routes to the BigIron RX's 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.

```
BigIron RX(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 these 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.

```
BigIron RX# 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*> specifies 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.

To resend routes to a neighbor without closing the neighbor session, enter a command such as the following.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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*> specifies 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

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 any level of the CLI.

```
BigIron RX# 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 848.

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 848.

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.

```
BigIron RX# 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.

```
BigIron RX# 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 827.

This information can be useful if you are working with Brocade 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 BGP 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.

```
BigIron RX# clear ip bgp neighbor 10.0.0.1 last-packet-with-error
BigIron RX# 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>* specifies 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.

Displaying BGP4 information

You can display the following configuration information and statistics for the BGP4 protocol on the router:

- Summary BGP4 configuration information for the router
- 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 router's BGP4 route table
- Route flap dampening statistics
- Active route maps (the route map configuration information in the running configuration)

Displaying summary BGP4 information

You can display the local AS number, the maximum number of routes and neighbors supported, and some BGP4 statistics.

To view summary BGP4 information for the router, enter the following command at any CLI prompt.

```
BigIron RX# show ip bgp summary
  BGP4 Summary
  Router ID: 101.0.0.1   Local AS Number : 4
  Confederation Identifier : not configured
  Confederation Peers: 4 5
  Maximum Number of Paths Supported for Load Sharing : 1
  Number of Neighbors Configured : 11, UP:2
  Number of Routes Installed : 2
  Number of Routes Advertising to All Neighbors : 8
  Number of Attribute Entries Installed : 6
  Neighbor Address  AS#   State   Time     Rt:Accepted Filtered Sent   ToSend
  1.2.3.4           200   ADMDN   0h44m56s    0        0        0      2
  10.0.0.2          5     ADMDN   0h44m56s    0        0        0      0
  10.1.0.2          5     ESTAB   0h44m56s    1        11       0      0
  10.2.0.2          5     ESTAB   0h44m55s    1        0        0      0
  10.3.0.2          5     ADMDN   0h25m28s    0        0        0      0
  10.4.0.2          5     ADMDN   0h25m31s    0        0        0      0
  10.5.0.2          5     CONN    0h 0m 8s    0        0        0      0
  10.7.0.2          5     ADMDN   0h44m56s    0        0        0      0
  100.0.0.1         4     ADMDN   0h44m56s    0        0        0      2
  102.0.0.1         4     ADMDN   0h44m56s    0        0        0      2
  150.150.150.150   0     ADMDN   0h44m56s    0        0        0      2
```

Syntax: show ip bgp summary

This display shows the following information.

TABLE 119 BGP4 summary information

This field...	Displays...
Router ID	The BigIron RX's router ID.
Local AS Number	The BGP4 AS number the router is in.
Confederation Identifier	The AS number of the confederation the BigIron RX is in.
Confederation Peers	The numbers of the local ASs contained in the confederation. This list matches the confederation peer list you configure on the BigIron RX.
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 "Changing the maximum number of shared BGP4 paths" on page 768.
Number of Neighbors Configured	The number of BGP4 neighbors configured on this BigIron RX, and currently in established state.
Number of Routes Installed	The number of BGP4 routes in the router's BGP4 route table. To display the BGP4 route table, refer to "Displaying the BGP4 route table" on page 839.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors.

TABLE 119 BGP4 summary information (Continued)

This field...	Displays...
Number of Attribute Entries Installed	The number of BGP4 route-attribute entries in the router's route-attributes table. To display the route-attribute table, refer to "Displaying BGP4 route-attribute entries" on page 846.
Neighbor Address	The IP addresses of this router's BGP4 neighbors.
AS#	The AS number.
State	The state of this router's neighbor session with each 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. Refer to "Administratively shutting down a session with a BGP4 neighbor" on page 779. • 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. **NOTE:** ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. 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 packets with the neighbor. **NOTE:** If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. 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 time that has passed since the state last changed.
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.

TABLE 119 BGP4 summary information (Continued)

This field...	Displays...
Sent	The number of BGP4 routes that the BigIron RX has sent to the neighbor.
ToSend	The number of routes the BigIron RX has queued to send to this 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.

```
BigIron RX# 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

To display summary neighbor information, enter a command such as the following at any level of the CLI.

```
BigIron RX(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 120 BGP4 route summary information for a neighbor

This field...	Displays...
IP Address	The IP address of the neighbor
Routes Received	How many routes the BigIron RX has received from the neighbor during the current BGP4 session. • Accepted/Installed – Indicates how many of the received routes the BigIron RX accepted and installed in the BGP4 route table. • Filtered/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 BigIron RX 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 BigIron RX 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 BigIron RX 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.

TABLE 120 BGP4 route summary information for a neighbor (Continued)

This field...	Displays...
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 BigIron RX has received. • Replacements – The number of replacement routes the BigIron RX has received.
NLRIs Discarded due to	Indicates the number of times the BigIron RX discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The BigIron RX'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 BigIron RX has advertised to this neighbor. • To be Sent – The number of routes the BigIron RX has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the BigIron RX has queued up to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the BigIron RX has sent to this neighbor in UPDATE messages. • Withdraws – The number of routes the BigIron RX has sent to the neighbor to withdraw. • Replacements – The number of routes the BigIron RX has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the BigIron RX 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 BGP4 neighbor information

You can display configuration information and statistics for the router's BGP4 neighbors.

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.

```
BigIron RX(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
       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. 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 BigIron RX's Transmission Control Block (TCB) for the TCP session between the device 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 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 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 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 815.

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 121 BGP4 neighbor information

This field...	Displays...
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.

TABLE 121 BGP4 neighbor information (Continued)

This field...	Displays...
EBGP/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 BigIron RX.
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. Refer to "Administratively shutting down a session with a BGP4 neighbor" on page 779. • 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. 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. **NOTE:** If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. 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. Refer to "Changing the keep alive time and hold time" on page 787.

TABLE 121 BGP4 neighbor information (Continued)

This field...	Displays...
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. Refer to "Changing the keep alive time and hold time" on page 787.
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 BigIron RX will accept from this neighbor.
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this BigIron RX 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 121 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 Brocade 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 121 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 121 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 BigIron RX.
Local port	The TCP port the BigIron RX 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 BigIron RX.
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 BigIron RX 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 BigIron RX 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.

TABLE 121 BGP4 neighbor information (Continued)

This field...	Displays...
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 route information for a neighbor

You can display routes based on the following criteria:

- A summary of the routes for a specific neighbor.
- The routes received from the neighbor that the device selected as the best routes to their destinations.
- 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 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.

```
BigIron RX(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 122 BGP4 route summary information for a neighbor

This field...	Displays...
Routes Received	How many routes the BigIron RX has received from the neighbor during the current BGP4 session. • Accepted/Installed – Indicates how many of the received routes the BigIron RX accepted and installed in the BGP4 route table. • Filtered – Indicates how many of the received routes the BigIron RX did not accept or install because they were denied by filters on the BigIron RX.
Routes Selected as BEST Routes	The number of routes that the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX has received. • Replacements – The number of replacement routes the BigIron RX has received.
NLRIs Discarded due to	Indicates the number of times the BigIron RX discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The BigIron RX'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 BigIron RX has advertised to this neighbor. • To be Sent – The number of routes the BigIron RX has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the BigIron RX has queued up to send to this neighbor in UPDATE messages.

TABLE 122 BGP4 route summary information for a neighbor (Continued)

This field...	Displays...
NLRIs Sent in Update Message	The number of NLRIs for new routes the BigIron RX has sent to this neighbor in UPDATE messages. • Withdraws – The number of routes the BigIron RX has sent to the neighbor to withdraw. • Replacements – The number of routes the BigIron RX has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the BigIron RX 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 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.

```
BigIron RX# 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, as in the following example.

```
BigIron RX# 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 124 on page 842. The fields in this display also appear in the **show ip bgp** display.

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.

```
BigIron RX(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 124 on page 842. The fields in this display also appear in the **show ip bgp** display.

Displaying the adj-RIB-out for a neighbor

To display the BigIron RX's 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.

```
BigIron RX(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 124 on page 842. The fields in this display also appear in the **show ip bgp** display.

Displaying peer group information

You can display configuration information for peer groups.

To display peer-group information, enter a command such as the following at the Privileged EXEC level of the CLI.

```
BigIron RX# 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 BigIron RX's BGP4 route table, enter a command such as the following at any level of the CLI.

```
BigIron RX(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 123 BGP4 summary route information

This field...	Displays...
Total number of BGP routes (NLRIs) Installed	The number of BGP4 routes the BigIron RX has installed in the BGP4 route table.
Distinct BGP destination networks	The number of destination networks the installed routes represent. The BGP4 route table can have multiple routes to the same network.
Filtered BGP routes for soft reconfig	The 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 815.
Routes originated by this router	The number of routes in the BGP4 route table that this BigIron RX originated.
Routes selected as BEST routes	The number of routes in the BGP4 route table that this BigIron RX has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	The number of BGP4 routes that are the best BGP4 routes to their destinations but were not installed in the IP route table because the BigIron RX received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable routes (no IGP route for NEXTHOP)	The number of routes in the BGP4 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 BGP4 route table that are IBGP routes.
EBGP routes selected as best routes	The number of "best" routes in the BGP4 route table that are EBGP routes.

Displaying the BGP4 route table

BGP4 uses filters you define as well as the algorithm described in "How BGP4 selects a path for a route" on page 740 to determine the preferred route to a destination. BGP4 sends only the preferred route to the router's IP table. However, if you want to view all the routes BGP4 knows about, you can display the BGP4 table.

To view the BGP4 route table, enter the following command.

```
BigIron RX(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 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. Refer to "Using regular expressions" on page 794.

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.

Displaying the best BGP4 routes

To display all the BGP4 routes in the BigIron RX'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.

```
BigIron RX(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 124 on page 842. 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.

```
BigIron RX(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 124 on page 842. 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.

```
BigIron RX(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 paths 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.

```
BigIron RX(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 124 BGP4 network information

This field...	Displays...
Number of BGP 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 router for reaching the network from the BigIron RX.
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 124 BGP4 network information (Continued)

This field...	Displays...
Weight	The value that this router 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 route's AS path. **NOTE:** This field appears only if you *do not* 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. **NOTE:** This field appears only if you *do not* enter the **route** option.
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. **NOTE:** B – BEST. BGP4 has determined that this is the optimal route to the destination. 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 BigIron RX 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 BigIron RX. **NOTE:** 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". If the "m" is shown in lowercase, the software was not able to install the route in the IP route table. **NOTE:** S – SUPPRESSED. This route was suppressed during This field appears only if you enter the **route** option.

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.

```
BigIron RX# 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 125 BGP4 route information

This field...	Displays...
Total number of BGP Routes	The number of BGP4 routes.
Status codes	A list of the characters the display uses to indicate the route's 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's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. **NOTE:** B – BEST. BGP4 has determined that this is the optimal route to the destination. 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 BigIron RX 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 BigIron RX. **NOTE:** 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". 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.

TABLE 125 BGP4 route information (Continued)

This field...	Displays...
Age	The last time an update occurred.
Next_Hop	The next-hop router for reaching the network from the BigIron RX.
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.
Weight	The value that this router 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.
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 BigIron RX 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 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 IP route table, enter the following command.

```
BigIron RX# show ip bgp attribute-entries
```

Syntax: show ip bgp attribute-entries

Here is an example of the information displayed by this command. A zero value indicates that the attribute is not set.

```
BigIron RX# 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 126 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.

TABLE 126 BGP4 route-attribute entries information (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 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.

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. You can view the IP route table.

To display the IP route table, enter the following command.

```
BigIron RX# show ip route
```

Syntax: show ip route [<*ip-addr*> | <*num*> | bgp | ospf | rip | isis]

Here is an example of the information displayed by this command. Notice that most of the routes in this example have type "B", indicating that their source is BGP4.

```
BigIron RX# 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.

```
BigIron RX# 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 794.

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 **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 **filter-list** <*num*> parameter specifies one or more filters. Only the routes that have been dampened and that match the specified filters are displayed.

This display shows the following information.

TABLE 127 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the BigIron RX'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 neighbor that sent the route to the BigIron RX.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time since the first flap of this route.

TABLE 127 Route flap dampening statistics

This field...	Displays...
Reuse	The amount of time remaining until this route will be un-suppressed and thus be usable again.
Path	Shows the AS-path information for the route.

You also can display all the dampened routes by entering the following command.
show ip bgp dampened-paths.

Displaying the active route map configuration

You can view the device's active route map configuration (contained in the running configuration) without displaying the entire running configuration.

To display the device's active route map configuration, enter the following command at any level of the CLI.

```
BigIron RX# 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.

```
BigIron RX# 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 called "setcomm".

Syntax: show route-map [<*map-name*>]

Graceful restart in BGP

Under normal operation, restarting a BGP router causes the network to be reconfigured. In this situation, routes available through the restarting router are first deleted when the router goes down and are then rediscovered and re-added to the routing tables when the router is back up and running. In a network where routers are restarted regularly, this can degrade performance

significantly and limit the availability of network resources. BGP graceful restart dampens the network topology changes and limits route flapping by allowing routes to remain available between routers during a restart. BGP Graceful restart operates between a router and its peers and must be configured on both the router and its peers.

A BGP router with graceful restart enabled advertises its graceful restart capability and restart timer to establish peering relationships with other routers. Once the restarting router is restarted, it begins to reestablish BGP connections and receive routing updates from its peers. When the restarting router receives all end-of-RIB markers from its helper neighbors, all of the routes are recomputed, and newly computed routes replace the stale routes in the routing table. An end-of-RIB marker indicates that it has received all of the BGP route updates.

During the restarting process, the helper neighbors will continue to use all of the routes learned from the restarting router and mark them as stale for the length of the learned restart timer. If the restarting router doesn't come back up within the restart timer, the routes marked stale will be removed.

Configuring BGP graceful restart

To configure BGP Graceful Restart, you must enable it on all BGP peers where you want it to operate and set the following timers:

- Restart Timer
- Stale Routes Timer

NOTE
After configuring BGP Graceful Restart, you need to reset neighbor session whether or not the neighbor session is up to enable BGP graceful restart. Use the **clear ip bgp neighbor** command to clear and re-establish neighbor sessions.

Configuring BGP graceful restart on a router

Use the following command to enable the BGP graceful restart feature on a BigIron RX Switch.

```
BigIron RX(config)#router bgp
BigIron RX(config-bgp)#graceful-restart
```

Configuring BGP graceful restart timer

Use the following command to specify the maximum amount of time a router will maintain routes from a restarting router and forward traffic to a restarting router.

```
BigIron RX(config)#router bgp
BigIron RX(config-bgp)#graceful-restart
BigIron RX(config-bgp)#graceful-restart restart-time 60
```

Syntax: graceful-restart restart-time <*seconds*>

The <*seconds*> variable sets the maximum number of seconds the restarting router will take to restart. Also, the peer routers waits this number of seconds to re-establish BGP connection and to keep using the learned routes from the restarting router. Enter 10 – 3600 seconds. The default value is 120 seconds.

Configuring BGP graceful restart stale routes timer

Use the following command to specify the maximum amount of time a helper router will wait for an end-of-RIB message from a restarting router before deleting stale routes learned from that restarting router.

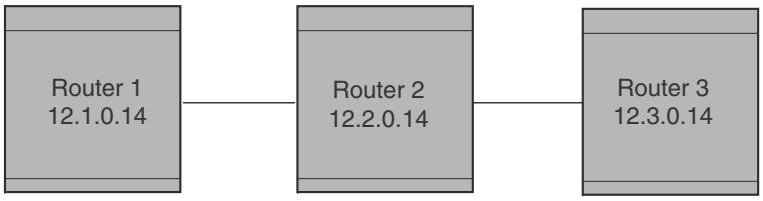
```
BigIron RX(config-bgp)#graceful-restart stale-routes-time 30
```

Syntax: graceful-restart stale-routes-time <*seconds*>

The <*seconds*> variable sets the number of seconds that a helper router will wait for an end-of-RIB (restart complete) message from a restarting router. Enter 10 – 3600 seconds. The default value is 360 seconds.

BGP graceful restart example

The following example configures three routers for BGP Graceful Restart. The default timer values are accepted in this configuration.



Restarting Router

Router 1

```
BigIron RX(config)#router bgp
BigIron RX(config-bgp)#local-as 100
BigIron RX(config-bgp)#graceful-restart
BigIron RX(config-bgp)#neighbor 12.2.0.14 remote-as 200
BigIron RX(config-bgp)#write memory
```

Router 2

```
BigIron RX(config)#router bgp
BigIron RX(config-bgp)#local-as 200
BigIron RX(config-bgp)#graceful-restart
BigIron RX(config-bgp)#neighbor 12.1.0.14 remote-as 100
BigIron RX(config-bgp)#neighbor 12.3.0.14 remote-as 300
BigIron RX(config-bgp)#write memory
```

Router 3

```
BigIron RX(config)#router bgp
BigIron RX(config-bgp)#local-as 300
BigIron RX(config-bgp)#graceful-restart
BigIron RX(config-bgp)#neighbor 12.2.0.14 remote-as 100
BigIron RX(config-bgp)#write memory
```

Displaying BGP graceful restart information

You can display the BGP Graceful Restart configuration by entering the following command.

```
BigIron RX# show ip bgp neighbor 11.11.11.2
1 IP Address: 11.11.11.2, Remote AS: 101 (EBGP), RouterID: 101.101.101.1
Local AS: 200
State: ESTABLISHED, Time: 0h18m15s, KeepAliveTime: 60, HoldTime: 180
KeepAliveTimer Expire in 44 seconds, HoldTimer Expire in 167 seconds
RefreshCapability: Received
GracefulRestartCapability: Received
Restart Time 120 sec, Restart bit 0
afi/safi 1/1, Forwarding bit 0
GracefulRestartCapability: Sent
Restart Time 30 sec, Restart bit 0
afi/safi 1/1, Forwarding bit 0
Messages: Open Update KeepAlive Notification Refresh-Req
Sent : 1 5 15 0 0
Received: 1 1 15 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: 628, Received: 363
Local host: 11.11.11.1, Local Port: 8190
Remote host: 11.11.11.2, Remote Port: 179
ISentSeq: 2123652 SendNext: 2124281 TotUnAck: 0
TotSent: 629 ReTrans: 1 UnAckSeq: 2124281
IRcvSeq: 2300094 RcvNext: 2300458 SendWnd: 65000
TotalRcv: 364 DupliRcv: 0 RcvWnd: 65000
SendQue: 0 RcvQue: 0 CngstWnd: 1460
```

Syntax: show ip bgp neighbor <*address*>

Generalized TTL security mechanism support

The device supports the Generalized TTL Security Mechanism (GTSM) as defined in RFC 3682. GTSM provides a means of protecting the Brocade device from attacks where invalid BGP control traffic is sent to the device in order to overload the CPU or hijack the BGP session. GTSM protection applies to EBGP neighbors only.

When GTSM protection is enabled, BGP control packets sent by the Brocade device to its neighbor have a Time To Live (TTL) value of 255. In addition, the Brocade device expects the BGP 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 Brocade device expects the TTL for BGP 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 BGP control packets received from the neighbor do not have the anticipated value, they are dropped by the Brocade device.

For more information on GTSM protection, see RFC 3682.

To enable GTSM protection for neighbor 192.168.9.210, enter the following command.

```
BigIron RX(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 Brocade device and the neighbor.

Chapter

27

Configuring MBGP

In this chapter

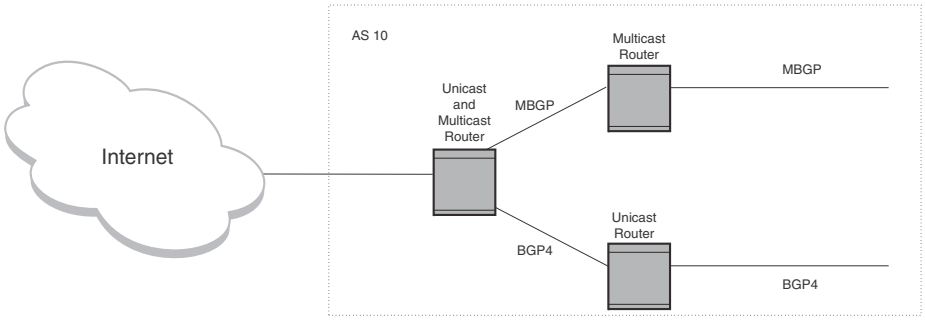
This chapter provides details on how to configure ***Multi-protocol Border Gateway Protocol (MBGP)***. MBGP is an extension to BGP that allows a router to support separate unicast and multicast topologies. BGP4 cannot support a multicast network topology that differs from the network's unicast topology. 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.

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 116 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 116 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 Brocade 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

- MBGP does not redistribute DVMRP routes. It redistributes static routes only.
- You cannot redistribute MBGP routes into BGP4.
- The device supports 8192 multicast routes by default. You may need to increase the maximum number of multicast routes for MBGP. You can configure the device to support up to 153,600 multicast routes.

Configuring MBGP

1. Optional – Set the maximum number of multicast routes supported by the device.
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 device in order for the device to send multicast prefixes to other multicast routers.
   - 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

The device supports up 1024 – 153,600 multicast routes.

NOTE
This procedure requires a software reload to place the change into effect.

To increase the maximum number of multicast routes supported on the device, enter commands such as the following.

```
BigIron RX(config)# system-max multicast-route 12000
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# 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 and can be from 1024 – 153,600.

Enabling MBGP

To enable MBGP4, you must enable PIM SM or DM and BGP4. Enter commands such as the following.

```
BigIron RX> enable
BigIron RX# configure terminal
BigIron RX(config)# router pim
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-1/1)# ip address 1.1.1.1/24
BigIron RX(config-if-1/1)# ip pim
BigIron RX(config-if-1/1)# exit
BigIron RX(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
BigIron RX(config-bgp)# local-as 10
```

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.

```
BigIron RX(config-bgp)#address-family ipv4 multicast
BigIron RX(config-bgp-ipv4m)#
```

Syntax: address-family ipv4 multicast

Adding MBGP neighbors

To add an MBGP neighbor, enter a command such as the following.

```
BigIron RX(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 device will exchange only multicast routes with the neighbor.

NOTE
If the BigIron RX 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*>
[advertisement-interval <*num*>]
[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]
[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.

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 BigIron RX 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 BigIron RX 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. For information on these other tasks, refer to Chapter 26, "Configuring BGP4 (IPv4 and IPv6)".

Advertising routes from the local AS to MBGP

You can configure the device 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 device 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 BigIron RX can create a local MBGP route.

To configure the device to advertise network 207.95.22.0/24 as a multicast route, enter the following command.

```
BigIron RX(config-bgp-ipv4m)# network 207.95.22.0 255.255.255.0
```

Syntax: 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.

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. Here is an example.

```
BigIron RX(config)# access-list 10 permit 207.95.22.0 0.0.0.255
BigIron RX(config)# route-map mbgpmap permit 1
BigIron RX(config-routemap mbgpmap)# match ip address 10
BigIron RX(config-routemap mbgpmap)# exit
BigIron RX(config)# router bgp
BigIron RX(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 device 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.

Configuring static IP multicast routes

To configure static IP multicast routes, enter commands such as the following.

```
BigIron RX(config)# ip mroute 207.95.10.0 255.255.255.0 interface ethernet 1/2
BigIron RX(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 device 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 device 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*> | null0] [<*cost*>] [distance < *num*>]

The **ip-addr** and **ip-mask** parameters specifies the PIM source for the route.

The **ethernet** <*slot/port*> parameter specifies a physical port.

The **ve** <*num*> parameter specifies a virtual interface.

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 BigIron RX always prefers directly connected routes over other routes.

Aggregating routes advertised to BGP4 neighbors

By default, the device advertises individual MBGP routes for all the multicast networks. The aggregation feature allows you to configure the device to aggregate routes in a range of networks into a single CIDR number. For example, without aggregation, the device 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 device 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.

```
BigIron RX(config-bgp-router)# aggregate-address 207.95.0.0 255.255.0.0
```

Syntax: 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.

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 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.

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 128 lists the MBGP show commands and describes their output. For information about a command, refer to Chapter 26, "Configuring BGP4 (IPv4 and IPv6)".

TABLE 128 MBGP Show commands

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.

TABLE 128 MBGP Show commands (Continued)

Command	Description
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.

The following sections show examples of some of the MBGP show commands. An example of the **show ip mroute** command is also included. This command displays the IP multicast route table.

Displaying summary MBGP information

To display summary MBGP information, enter the following command at any CLI prompt.

```
BigIron RX# 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.

Displaying the active MBGP configuration

To display the active MBGP configuration information contained in the running-config without displaying the entire running-config, enter the following command at any level of the CLI.

```
BigIron RX# 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.

Displaying MBGP neighbors

To view MBGP neighbor information including the values for all the configured parameters, enter the following 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.

```
BigIron RX # 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
```

This example shows 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 indicates the neighbor for which 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 BigIron RX 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 BigIron RX's Transmission Control Block (TCB) for the TCP session between the device and its neighbor. These fields are described in detail in section 3.2 of RFC 793, "Transmission Control Protocol Functional Specification".

Syntax: show ip mbgp neighbors [<*ip-addr*>]

The <*ip-addr*> parameter specifies the neighbor's IP address.

Displaying MBGP routes

To display the MBGP route table, enter the following command.

```
BigIron RX#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

Displaying the IP multicast route table

To display the IP multicast route table, enter the following command.

```
BigIron RX#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 IP multicast route information for a specific destination address only.

The **bgp** parameter displays IP multicast route information for BGP routes only.

The **static** parameter displays IP multicast route information for static routes only.

Chapter

28

Configuring Secure Shell

In this chapter

Overview of Secure Shell (SSH)

Secure Shell (SSH) is a mechanism for allowing secure remote access to management functions on a BigIron RX. 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 device. Brocade'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 device negotiates the version of SSHv2 to be used. The highest version of SSHv2 supported by both the device 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.

Also, BigIron RX support Secure Copy (SCP) for securely transferring files between a BigIron RX and an SCP-enabled remote hosts. Refer to "Using secure copy" on page 876 for more information.

NOTE
The SSH feature includes software that is copyright Allegro Software Development Corporation.

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/SFTP/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
- 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/IP port forwarding, X11 forwarding, and secure file transfer are not supported.
- SSH version 1 is not supported.
- SCP supports AES encryption

Configuring SSH

Brocade'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/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 BigIron RX, do the following.

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 device. The SSH server on the device 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 BigIron RX'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 BigIron RX's public key in it. Refer to "Providing the public key to clients" on page 870 for an example of what to place in the known hosts file.

While the SSH listener exists at all times, sessions can not 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 BigIron RX, enter the following commands.

```
BigIron RX(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 BigIron RX, enter the following commands.

```
BigIron RX(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 device to display the DSA host key pair in the running configuration file entering the following command.

```
BigIron RX# ssh show-host-keys
```

Syntax: ssh show-host-keys

To hide the public keys in the running configuration file, enter the following command.

```
BigIron RX# 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 BigIron RX from a UNIX system, you may need to add the device'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 device. 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 BigIron RX.
2. The device compares the client's public key to those stored in memory.
3. If there is a match, the device uses the public key to encrypt a random sequence of bytes.
4. The device 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 BigIron RX.

7. The device 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 device.
2. Enabling DSA challenge response authentication

Importing authorized public keys into the BigIron RX

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 device and place all of these keys into one file. This public key file is imported into the device.

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 ----
```

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.

NOTE
You must ensure the format be followed before the key is TFTPed to the Brocade device.

NOTE
The public key may not be effective after download using Linux and Secure CRT. If the file is not constructed properly, you will receive an error message while loading. You must fix the key files and load them again.

To cause a public key file called pkeys.txt to be loaded from a TFTP server each time the device is booted, enter a command such as the following.

```
BigIron RX(config)# ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
```

Syntax: ip ssh pub-key-file tftp | <*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 Brocade device.

The <*filename*> variable is the name of the dsa public key file that you want to import into the Brocade device.

The **remove** parameter deletes the key from the system.

To display the currently loaded public keys, enter the following command.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX(config)# ip ssh key-authentication yes
```

To disable DSA challenge-response authentication.

```
BigIron RX(config)# ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

Setting the number of SSH authentication retries

By default, the device 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.

```
BigIron RX(config)# ip ssh authentication-retries 5
```

Syntax: ip ssh authentication-retries <*number*>

Deactivating user authentication

After the SSH server on the device negotiates a session key and encryption method with the connecting client, user authentication takes place. Brocade'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 device. 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 873). 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.

```
BigIron RX(config)# ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

The default is "yes".

To deactivate password authentication.

```
BigIron RX(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 75 for information on setting up user names and passwords on the device.

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.

```
BigIron RX(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.

```
BigIron RX(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, Brocade recommends 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.

```
BigIron RX(config)# ip ssh timeout 60
```

Syntax: ip ssh timeout <*seconds*>

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 BigIron RX.

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.

```
BigIron RX(config)# int loopback 2
BigIron RX(config-lbif-2)# ip address 10.0.0.2/24
BigIron RX(config-lbif-2)# exit
BigIron RX(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 device.

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. For example.

```
BigIron RX(config)# interface ethernet 1/4
BigIron RX(config-if-e10000-1/4)# ip address 209.157.22.110/24
BigIron RX(config-if-e10000-1/4)# exit
BigIron RX(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 device closes it. For example, to set the maximum idle time for SSH sessions to 30 minutes.

```
BigIron RX(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 device 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.

Filtering SSH access using ACLs

You can permit or deny SSH access to the device using ACLs. To use ACLs, first create the ACLs you want to use. You can specify a numbered standard IPv4 ACL, a named standard IPv4 ACL.

Then enter the following command.

```
BigIron RX(config)# access-list 10 permit host 192.168.144.241
BigIron RX(config)# access-list 10 deny host 192.168.144.242 log
BigIron RX(config)# access-list 10 permit host 192.168.144.243
BigIron RX(config)# access-list 10 deny any
BigIron RX(config)# ssh access-group 10
```

Syntax: ssh access-group < *standard-named-acl*> | < *standard-numbered-acl*>

Refer to the section Chapter 21, “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 device device. You can disable 3-DES by entering the following command.

```
BigIron RX(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 device. To display information about SSH connections, enter the following command.

```
BigIron RX# show ip ssh
Connection Version  Encryption  Username
1          SSH-2    3des-cbc    Hanuma
2          SSH-2    aes128-cbc  Mikaila
3          SSH-2    aes192-cbc  Jenny
4          SSH-2    aes256-cbc  Mariah
5          SSH-2    3des-cbc    Logan
```

Syntax: show ip ssh [| begin < *expression*> | exclude < *expression*> | include < *expression*>]

This display shows the following information about the active SSH connections.

TABLE 129 SSH connection information

This field...	Displays...
Connection	The SSH connection ID. This can be from 1 – 5.
Version	The SSH version number. This should always be 1.5.

TABLE 129 SSH connection information (Continued)

This field...	Displays...
Encryption	The encryption method used for the connection.
Username	The user name for the connection.

The **show who** command also displays information about SSH connections. For example.

```
BigIron RX#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 192.168.144.241, user is hanuma
1 minutes 16 seconds in idle
2 established, client ip address 192.168.144.241, user is Mikaila
you are connecting to this session
18 seconds in idle
3 established, client ip address 192.168.144.241, user is Jenny
1 minutes 39 seconds in idle
4 established, client ip address 192.168.144.242, user is Mariah
41 seconds in idle
5 established, client ip address 192.168.144.241, user is Logan
23 seconds in idle
```

Syntax: show who [| begin< *expression*> | exclude< *expression*> | include< *expression*>]

To terminate one of the active SSH connections, enter the following command.

```
BigIron RX# 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) or FTP. 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 device, including the startup configuration and running configuration files, to or from an SCP-enabled remote host.

SCP is enabled by default and can be disabled. To disable SCP, enter the following command.

```
BigIron RX(config)# ip ssh scp disable
```

Syntax: ip ssh scp disable | enable

NOTE
If you disable SSH, SCP is also disabled.

The following are examples of using SCP to transfer files from and to a BigIron RX.

NOTE
When using SCP, you enter the **scp** commands on the SCP-enabled client, rather than the console on the BigIron RX.

NOTE
Certain SCP client options, including -p and -r, are ignored by the SCP server on the BigIron RX. If an option is ignored, the client is notified.

To copy a configuration file (c:\cfg\foundry.cfg) to the running configuration file on a BigIron RX at 192.168.1.50 and log in as user terry, enter the following command on the SCP-enabled client.

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:runConfig
```

If password authentication is enabled for SSH, the user is prompted for user terry's password before the file transfer takes place.

To copy the configuration file to the startup configuration file.

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:startConfig
```

To copy the configuration file to a file called config1.cfg on the PCMCIA flash card in slot 1 on a management module.

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:slot1:/config1.cfg
```

To copy the configuration file to a file called config1.cfg on the PCMCIA flash card in slot 2 on a management module.

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:slot2:/config1.cfg
```

To copy the running configuration file on a BigIron RX to a file called c:\cfg\fdryrun.cfg on the SCP-enabled client.

```
C:\> scp terry@192.168.1.50:runConfig c:\cfg\fdryrun.cfg
```

To copy the startup configuration file on a BigIron RX to a file called c:\cfg\fdrystart.cfg on the SCP-enabled client.

```
C:\> scp terry@192.168.1.50:startConfig c:\cfg\fdrystart.cfg
```

Chapter

29

Configuring IS-IS (IPv4)

In this chapter

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 Brocade 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
- 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 TLV (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 Brocade's implementation of IS-IS.

NOTE
The BigIron RX does not support routing of Connectionless-Mode Network Protocol (CLNP) packets. The BigIron RX uses IS-IS for TCP/IP only.

Relationship to IP route table

The IS-IS protocol has the same relationship to the device'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

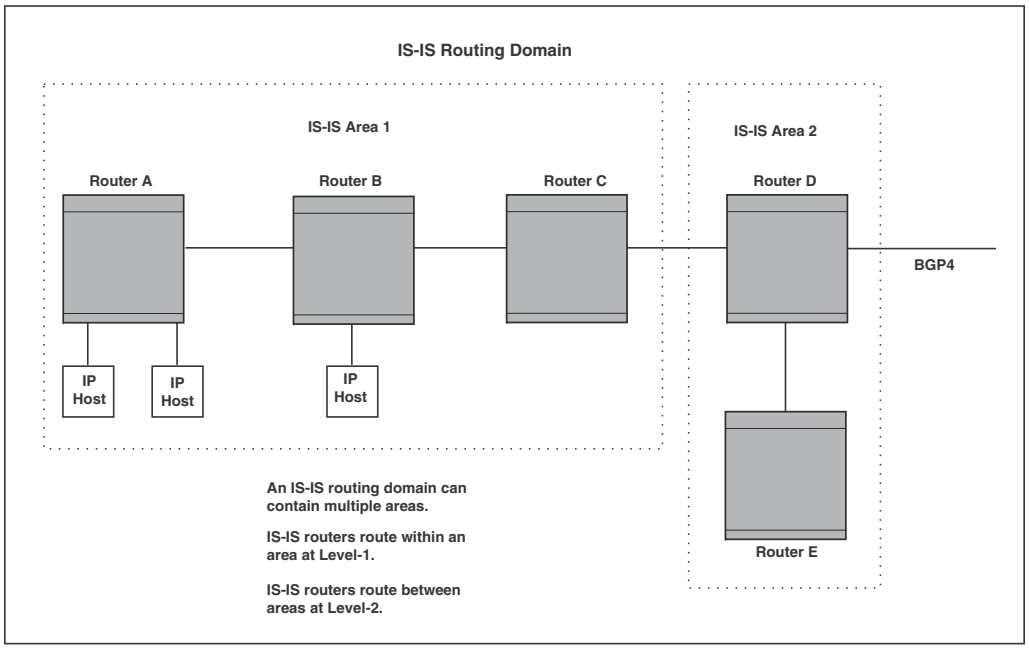
IS-IS uses the following categories to describe devices within an IS-IS routing domain (similar to an OSPF Autonomous System):

- ***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 BigIron RX, the device is an IS.

Figure 117 shows an example of an IS-IS network.

FIGURE 117 An IS-IS network contains Intermediate Systems (ISs) and host systems



NOTE
Since the Brocade 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 BigIron RX can be a member of one area for each Network Entity Title (NET) you configure on the device. The NET contains the area ID for the area the NET is in.

In Figure 117, 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 BigIron RX to perform one or both of the following levels of IS-IS routing[1]:

- **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.

1. The ISO/IEC specifications use the spelling “routeing”, but this document uses the spelling “routing” to remain consistent with other Brocade documentation.

In Figure 117 on page 881, 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 BigIron RX 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 device forms adjacencies are its ***neighbors***, which are other ISs.

BigIron RX IS-IS interfaces are configured by default for broadcast circuits.

In Figure 117 on page 881, 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 BigIron RX 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.

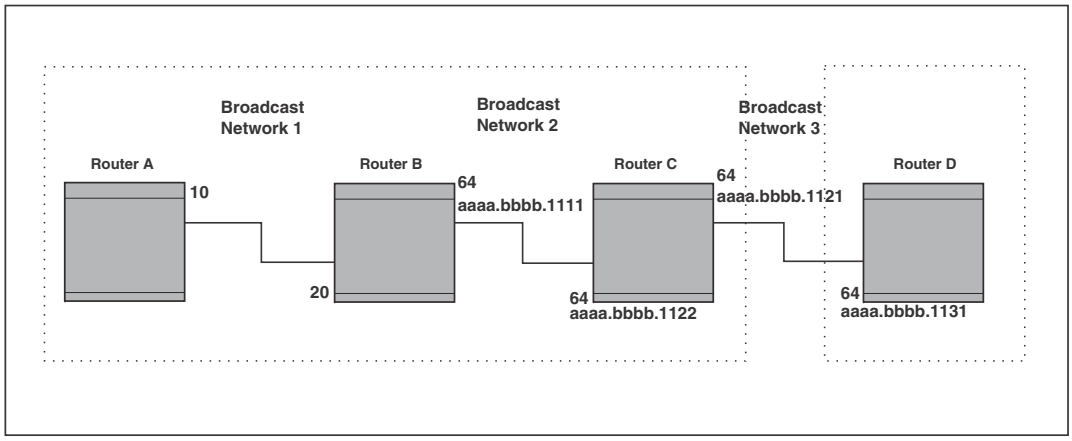
- The Level-1 IS that has the highest priority becomes the Level-1 Designated IS for the broadcast network.
- The Level-2 IS that has the highest priority becomes the Level-2 Designated IS for the broadcast network.

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 118 shows an example of the results of Designated IS elections. For simplicity, this example shows four of the five routers in Figure 117 on page 881, with the same domain and areas.

FIGURE 118 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.

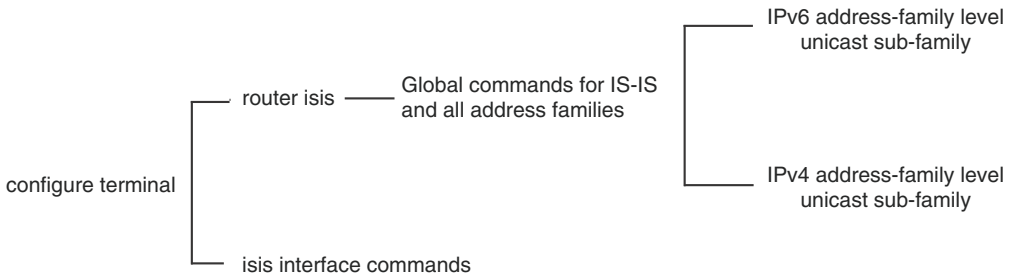
1. Prefer the Level-1 path over the Level-2 path.
2. If there is no Level-1 path, prefer the internal Level-2 path over the external Level-2 path.
3. If there is still more than one path, prefer the path with the lowest metric.
4. If there is more than one path with the lowest metric, load share among the paths.

After selecting the best path to a destination, the software places the path in the IS-IS route table.

IS-IS CLI levels

The CLI includes various levels of commands for IS-IS. Figure 119 diagrams these levels.

FIGURE 119 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**.

NOTE
IS-IS IPv6 is currently not supported.

- An interface level.

Global configuration level

You enter the global configuration level of ISIS by entering the following command.

```
BigIron RX(config)#router isis
BigIron RX(config-isis-router)#
```

Syntax: [no] 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 device 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, Brocade currently supports the unicast address family configuration level only. The device enters the IPv4 IS-IS unicast address family configuration level when you enter the following command while at the global IS-IS configuration level.

```
BigIron RX(config-isis-router)# address-family ipv4 unicast
BigIron RX(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.

```
BigIron RX(config-isis-router-ipv4u)# exit-address-family
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 2/3
BigIron RX(config-if-e1000-2/3)#ip router isis
```

Syntax: [no] ip router isis

Configuring IPv4 IS-IS

Enabling IS-IS globally

To configure IPv4 IS-IS, do the following.

1. Globally enable IS-IS by entering the following command.

   ```
   BigIron RX(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.

```
BigIron RX(config-isis-router)# net 49.2211.aaaa.bbbb.cccc.00
BigIron RX(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.

3. Configure ISIS parameters. Refer to "Globally configuring IS-IS on a device" on page 886, "Configuring IPv4 address family route parameters" on page 892, and "Configuring ISIS properties on an interface" on page 899.

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, although IPv6 is currently not supported.

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.

- 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 device 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 device from the network.

In addition, you can configure the device 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.

```
BigIron RX(config-isis-router)# set-overload-bit
```

This command administratively shuts down IS-IS by configuring the device to immediately set the overload bit to 1 (on) in all LSPs sent to other ISs.

To configure the device to temporarily set the overload bit on after a software reload, enter a command such as the following.

```
BigIron RX(config-isis-router)# set-overload-bit on-startup 5
```

This command configures the device 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 device 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 0 or a number from 5 – 86400 (24 hours). The default is 0, which means the device starts performing IS-IS routing immediately following a successful software reload.

Configuring authentication

By default, the device does not authenticate packets sent to or received from ESs or other ISs. You can configure the following types of passwords for IS-IS globally.

TABLE 130 IS-IS passwords

Password type	Scope	Where used	Default
Domain	Level-2	Level-2 LSPDU	None configured
Area	Level-1	Level-1 LSPDU	None configured
Interface	Level-1 and Level-2	Hello PDU	None configured

If you configure a password, the device checks for the password in IS-IS packets received by the device and includes the password in packets sent by the device. For example, the device checks all Level-2 LSPDUs received by the device for the domain password you configure, and includes the password in all Level-2 PDUs sent by the device.

Configuring a domain password

To configure an IS-IS domain password, enter a command such as the following.

```
BigIron RX(config-isis-router)# domain-password domain-1
```

This command configures the device to use the password “domain-1” to authenticate Level-2 LSPDUs.

Syntax: [no] domain-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, **domain-password “domain 1”**.

Configuring an area password

To configure an IS-IS area password, enter a command such as the following.

```
BigIron RX(config-isis-router)# area-password area-51
```

This command configures the device to use the password “area-51” to authenticate Level-1 LSPDUs.

Syntax: [no] area-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, **area-password “area 51”.**

Changing the IS-IS Level globally

By default, a BigIron RX 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.

```
BigIron RX(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 900.

Disabling or re-enabling display of hostname

Brocade'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 device to "IS-IS Router 1", the mapping feature uses this name instead of the BigIron RX's IS-IS system ID in the output of the following commands.

- show isis database
- show isis interface
- show isis neighbor

The BigIron RX's hostname is displayed in each CLI command prompt, for example.

```
BigIron RX(config-isis-router)#
```

The name mapping feature is enabled by default. If you want to disable name mapping, enter the following command.

```
BigIron RX(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 device 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.

```
BigIron RX(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
Although the command name is **csnp-interval**, the interval also applies to PSNPs.

Changing the maximum LSP lifetime

The maximum LSP lifetime is the maximum number of seconds an un-refreshed LSP can remain in the BigIron RX'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.

```
BigIron RX(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 BigIron RX's originated LSPs may be timed out by it's neighbors. Refer to "Changing the LSP refresh interval" on page 890.

Changing the LSP refresh interval

The LSP refresh interval is the maximum number of seconds the device 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.

```
BigIron RX(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 device 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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config-isis-router)# etransmit-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 device recalculates its IS-IS tree every five seconds following a change. You can change the SPF timer to a value from 1 – 120 seconds.

To change the SPF interval, enter a command such as the following.

```
BigIron RX(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.

Globally disabling or re-enabling hello padding

By default, the device adds extra data to the end of a hello packet to make the packet the same size as the maximum length of PDU the device 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 device can receive. Other ISs that receive a padded hello PDU from the device can therefore ensure that the IS-IS PDUs they send the device. Similarly, if the device receives a padded hello PDU from a neighbor IS, the device knows the maximum size PDU that the `device` can send to the neighbor.

When padding is enabled, the maximum length of a Hello PDU sent by the `device` 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.

```
BigIron RX(config-isis-router)# no hello padding
```

This command disables all hello PDU padding on the device. To re-enable padding, enter the following command.

```
BigIron RX(config-isis-router)# hello padding
```

Syntax: [no] hello padding

By default, hello padding is enabled. Enter the **no** form of the command to disable hello padding.

To disable hello padding on an interface, refer to "Disabling and enabling hello padding on an interface" on page 901.

Logging adjacency changes

The device can generate a Syslog entry and an SNMP trap to indicate a change 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.

```
BigIron RX(config-isis-router)# log-adjacency-changes
```

Syntax: [no] log-adjacency-changes

To disable logging of adjacency changes, enter the following command.

```
BigIron RX(config-isis-router)# no log-adjacency-changes
```

Disabling partial SPF calculations

By default, IS-IS makes incremental changes to the routing table when changes to the network occur. A full SPF calculation is not performed unless there is a substantial change in the network; for example when an IS-IS link flaps in the network. You can optionally configure IS-IS to perform a full SPF calculation when any changes occur in the network.

To disable partial SPF calculations for IS-IS, enter the following command.

```
BigIron RX(config-isis-router)# disable-partial-spf-opt
```

Syntax: [no] disable-partial-spf-opt

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 "Address family configuration level" on page 884.

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 device 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 device by changing the metric style to wide. The wide metric style allows metric values from 1 – 16777215.

To change the metric style to wide, enter the following command.

```
BigIron RX(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 device 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.

```
BigIron RX(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 device 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 `device` 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.

```
BigIron RX(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.

```
BigIron RX(config)# route-map default_level1 permit 1
BigIron RX(config-routemap default_level1)# set level level-1
BigIron RX(config-routemap default_level1)# exit
BigIron RX(config)# router isis
BigIron RX(config-isis-router)# address-family ipv4 unicast
BigIron RX(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).

Changing the administrative distance for IPv4 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 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 BigIron RX:

- 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 device 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.

```
BigIron RX(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 device 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.

```
BigIron RX(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 device 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 device 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 "Changing the default redistribution metric" on page 896 , 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 Brocade 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.

```
BigIron RX(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.

```
BigIron RX(config-isis-router-ipv4u)# redistribute static
```

This command configures the device to redistribute all static IPv4 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 IPv4 IS-IS level.

The **metric** <*number*> parameter restricts the redistribution to only those routes that have the metric you specify.

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.

```
BigIron RX(config)# access-list 101 permit ip any 192.168.0.0 255.255.0.0
BigIron RX(config)# route-map static permit 1
BigIron RX(config-routemap static)# match ip address 101
BigIron RX(config-routemap static)# router isis
BigIron RX(config-isis-router)# address-family ipv4 unicast
BigIron RX(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.

```
BigIron RX(config-isis-router-ipv4u)# redistribute connected
```

This command configures the device 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.

```
BigIron RX(config-isis-router-ipv4u)# redistribute rip
```

This command configures the device 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.

```
BigIron RX(config-isis-router-ipv4u)# redistribute ospf
```

This command configures the BigIron RX 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.

```
BigIron RX(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 device 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 BigIron RX 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.

```
BigIron RX(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. To configure an IP prefix list, refer to the *Enterprise Configuration and Management Guide*.

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.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-1/1)# ip router isis
BigIron RX(config-if-1/1)# exit
BigIron RX(config)# interface ethernet 1/2
BigIron RX(config-if-1/2)# ip router isis
```

These commands enable IS-IS on ports 1/1 and 1/2. 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 BigIron RX 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.

```
BigIron RX(config)# interface ethernet 2/8
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 2/8
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 2/8
BigIron RX(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 888 explains how to change the IS-IS level globally. By default, a BigIron RX 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.

```
BigIron RX(config)# interface ethernet 2/8
BigIron RX(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 891 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.

```
BigIron RX(config)# interface ethernet 2/8
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 2/8
BigIron RX(config-if-e1000-2/8)# isis hello-interval 20
```

This command changes the hello interval to 20 seconds. By default, the change applies to both 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 from
3 – 1000.

To change the hello multiplier for Ethernet interface 2/8, enter commands such as the following.

```
BigIron RX(config)# interface ethernet 2/8
BigIron RX(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 device originates an IS-IS route or calculates a route, the device adds a metric (cost) to the route. Each IS-IS interface has a separate metric value. The default is 10.

The device applies the interface-level metric to routes originated on the interface and also when calculating routes. The BigIron RX does not apply the metric to link-state information that the device 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.

```
BigIron RX(config)# interface ethernet 2/8
BigIron RX(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:

- **The active configuration** (the IS-IS commands in the running-config) – refer to "Displaying the IS-IS configuration in the running-config" on page 903
- **Name mappings** – "Displaying the name mappings" on page 903

- **Neighbor information** – "Displaying neighbor information" on page 904
- **Neighbor adjacency changes** – "Displaying IS-IS Syslog messages" on page 905
- **Interface information** – "Displaying interface information" on page 906
- **Route information** – "Displaying route information" on page 908
- **LSP database entries** – "Displaying LSP database entries" on page 909
- **Traffic statistics** – "Displaying traffic statistics" on page 912
- **Error statistics** – "Displaying error statistics" on page 913

Displaying the IS-IS configuration in the running-config

You can display the global IS-IS configuration commands that are in effect on the device 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 BigIron RX's running-config, enter the following command at any level of the CLI.

```
BigIron RX# 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.

```
BigIron RX# show isis hostname
Total number of entries in IS-IS Hostname Table: 1
   System ID        Hostname            * = local IS
 * bbbb.cccc.dddd    RX
```

Syntax: show isis hostname

The table in this example contains one mapping, for this device. The BigIron RX's IS-IS system ID is "bbbb.cccc.dddd" and its hostname is "RX". 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 889.

Displaying neighbor information

To display IS-IS neighbor information, enter the following command at any level of the CLI.

```
BigIron RX# 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 131 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	The number of ISs with which the BigIron RX has formed IS-IS adjacencies.
System ID	The System ID of the neighbor or the hostname of the neighbor.
Interface	The BigIron RX port or virtual interface attached to the neighbor.
SNPA	The Subnetwork Point of Attachment (SNPA), which is the MAC address of the BigIron RX 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 BigIron RX forms a separate adjacency for each IS-IS type. Thus, if the BigIron RX 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 device generates Syslog messages and SNMP traps for the following IS-IS events:

- Overload state (the device entering or leaving the overload state)
- Memory overrun (IS-IS is demanding more memory than is available)

You also can enable the device 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 892.

To display Syslog entries, enter the following command at any level of the CLI.

```
BigIron RX# 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 device and three ISs have come up.

Syntax: show logging

Table 132 lists the IS-IS Syslog messages.

TABLE 132 IS-IS Syslog messages

Message level	Message	Explanation
Alert	ISIS MEMORY USE EXCEEDED	IS-IS is requesting more memory than is available.
Notification	ISIS L1 ADJACENCY DOWN <*system-id*> on interface <*interface-id*>	The BigIron RX's adjacency with this Level-1 IS has gone down. The <*system-id*> is the system ID of the IS. The <*interface-id*> is the ID of the interface over which the adjacency was established.
Notification	ISIS L1 ADJACENCY UP <*system-id*> on interface <*interface-id*>	The BigIron RX's adjacency with this Level-1 IS has come up. The <*system-id*> is the system ID of the IS. The <*interface-id*> is the ID of the interface over which the adjacency was established.
Notification	ISIS L2 ADJACENCY DOWN <*system-id*> on interface <*interface-id*>	The BigIron RX's adjacency with this Level-2 IS has gone down. The <*system-id*> is the system ID of the IS. The <*interface-id*> is the ID of the interface over which the adjacency was established.

TABLE 132 IS-IS Syslog messages (Continued)

Message level	Message	Explanation
Notification	ISIS L2 ADJACENCY UP <*system-id*> on interface <*interface-id*>	The BigIron RX's adjacency with this Level-2 IS has come up. The <*system-id*> is the system ID of the IS. The <*interface-id*> is the ID of the interface over which the adjacency was established.
Notification	ISIS ENTERED INTO OVERLOAD STATE	The BigIron RX has set the overload bit to (1), indicating that the BigIron RX's IS-IS resources are overloaded.
Notification	ISIS EXITED FROM OVERLOAD STATE	The BigIron RX has set the overload bit to off (0), indicating that the BigIron RX's IS-IS resources are no longer overloaded.

Displaying interface information

To display information about the device's IS-IS interfaces, enter the following command at any level of the CLI.

```
BigIron RX# show isis interface

Total number of IS-IS Interfaces: 1

Interface: Eth 7/1
   Circuit State: UP Circuit Mode: LEVEL-1-2
   Circuit Type: BCAST Passive State: FALSE
   Circuit Number: 0x01, MTU: 1497
   Authentication password: None
   Level-1 Metric: 10, Level-1 Priority: 64
   Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
   Level-1 Designated IS: RX-01 Level-1 DIS Changes: 8
   Level-2 Metric: 10, Level-2 Priority: 64
   Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
   Level-2 Designated IS: RX-01 Level-2 DIS Changes: 8
   Next IS-IS LAN Level-1 Hello in 9 seconds
   Next IS-IS LAN Level-2 Hello in 5 seconds
   Number of active Level-1 adjacencies: 1
   Number of active Level-2 adjacencies: 1
   Circuit State Changes: 1 Circuit Adjacencies State Changes: 6
   Rejected Adjacencies: 0
   Circuit Authentication Fails: 0 Bad LSPs: 0
   Control Messages Sent: 602 Control Messages Received: 2212
   IP Enabled: TRUE
   IP Address and Subnet Mask:
     4.1.1.2             255.255.255.0
```

Syntax: show isis interface [brief | ethernet <*slot-number*>/<*port-number*> | loopback <*number*> | ve <*number*>]

This display shows the following information.

TABLE 133 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.
Local 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.
Circuit Type	The type of IS-IS circuit running on the interface. The circuit type is set to BCAST (broadcast).
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 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.
MTU	The maximum length supported for IS-IS PDUs sent on this interface.
Authentication Password	The password assigned to the IS-IS interface.
Level-1 Metric	The default-metric value that the BigIron RX 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.
Level-2 Metric	The default-metric value that the BigIron RX 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.

TABLE 133 IS-IS Interface information (Continued)

This field...	Displays...
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 BigIron RX.
Next IS-IS LAN Level-2 Hello	Number of seconds before next Level-2 Hello PDU will be transmitted by the BigIron RX.
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 BigIron RX.
Circuit Authentication Fails	The number of times the BigIron RX rejected a circuit because the authentication did not match the authentication configured on the BigIron RX.
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	If set to TRUE, the IP protocol is enabled for this circuit.

Displaying route information

To display the routes in the BigIron RX's IS-IS route table, use either of the following methods.

To display information about the routes in the BigIron RX's IS-IS route table, enter the following command at any level of the CLI.

```
BigIron RX# 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.

For example:

```
BigIron RX# 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 134 IS-IS route information

This field...	Displays...
Total number of IS-IS routes	The total number of routes in the BigIron RX'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.
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 BigIron RX 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 BigIron RX interface (port or virtual interface) attached to the next hop.
Flags	Values used by Brocade 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 BigIron RX 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 BigIron RX's LSP databases, enter the following command at any level of the CLI.

```
BigIron RX)# show isis database
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
RX-1.00-00           0x0000000c   0xd048        963               1/0/0
RX-1.01-00           0x00000004   0x09b0        957               0/0/0
RX-1.02-00           0x00000001   0xc57b        961               0/0/0
RX.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
RX-1.00-00           0x0000000d   0x7d97        964               1/0/0
RX-1.01-00           0x00000004   0x09b0        958               0/0/0
RX-1.02-00           0x00000001   0x200f        962               0/0/0
RX.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 BigIron RX'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, RX.00-00.

The **detail** parameter displays detailed information about the LSPs. Refer to "Displaying detailed information" on page 911.

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 135 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 BigIron RX 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 BigIron RX's LSP database.
ATT	A 4-bit value extracted from bits 4 – 7 in the Attach field of the LSP.

TABLE 135 IS-IS summary LSP database information (Continued)

This field...	Displays...
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 BigIron RX's LSP databases, enter the following command at any level of the CLI.

```
BigIron RX# show isis database detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
RX.00-00*            0x0000000b   0x23fb        971               1/0/0
   Area Address:  49
   NLPID:  CC(IP)
   Hostname:  RX
   Metric:  10    IP-Internal 4.1.1.0/24          Up-bit: 0
   Metric:  10    IS RX.01

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
RX.00-00*            0x0000000d   0x7d97        903               1/0/0
   Area Address:  49
   NLPID:  CC(IP)
   Hostname:  RX
   IP address:  4.1.1.1
   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 RX.01
   Metric:  10    IS RX.02
```

TABLE 136 IS-IS detailed LSP database information

This field...	Displays...
LSPID	See the description of the summary display.
LSP Seq Num	See the description of the summary display.
LSP Checksum	See the description of the summary display.
LSP Holdtime	See the description of the summary display.
ATT/P/OL	See 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)".

TABLE 136 IS-IS detailed LSP database information (Continued)

This field...	Displays...
IP address	The IP address of the interface that sent the LSP. The BigIron RX can use this address as the next hop in routes to the addresses listed in the rows below.
Destination addresses	The rows of information below the IP address row are the destinations advertised by the LSP. The BigIron RX 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.

Displaying traffic statistics

The BigIron RX 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.

```
BigIron RX# 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 137 IS-IS traffic statistics

This field...	Displays...
Level-1 Hellos	The number of Level-1 hello PDUs sent and received by the BigIron RX.
Level-2 Hellos	The number of Level-2 hello PDUs sent and received by the BigIron RX.
Level-1 LSP	The number of Level-1 link-state PDUs sent and received by the BigIron RX.

TABLE 137 IS-IS traffic statistics (Continued)

This field...	Displays...
Level-2 LSP	The number of Level-2 link-state PDUs sent and received by the BigIron RX.
Level-1 CSNP	The number of Level-1 Complete Sequence Number PDUs (CSNPs) sent and received by the BigIron RX.
Level-2 CSNP	The number of Level-2 CSNPs sent and received by the BigIron RX.
Level-1 PSNP	The number of Level-1 Partial Sequence Number PDUs (PSNPs) sent and received by the BigIron RX.
Level-2 PSNP	The number of Level-2 PSNPs sent and received by the BigIron RX.

Displaying error statistics

To display IS-IS error statistics, enter the following command at any level of the CLI.

```
BigIron RX# show 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 isis counts

This display shows the following information.

TABLE 138 IS-IS error statistics

This field...	Displays...
Area Mismatch	The number of times the BigIron RX interface was unable to create a Level-1 adjacency with a neighbor because the BigIron RX interface and the neighbor did not have any areas in common.
Max Area Mismatch	The number of times the BigIron RX received a PDU whose value for maximum number of area addresses did not match the BigIron RX's value for maximum number of area addresses.
System ID Length Mismatch	The number of times the BigIron RX received a PDU whose ID field was a different length than the ID field length configured on the BigIron RX.
Authentication Fail	The BigIron RX is configured to authenticate 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 BigIron RX detected a corrupted LSP in the device's memory.
LSP Sequence Number Skipped	The number of times the BigIron RX 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 BigIron RX attempted to set an LSP sequence number to a value higher than the highest number in the CSNP sent by the Designated IS.

TABLE 138 IS-IS error statistics (Continued)

This field...	Displays...
Level-1 Database Overload	The number of times the Level-1 state on the BigIron RX changed from Waiting to On or from On to Waiting. • **Waiting to On** – This change can occur when the BigIron RX 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 BigIron RX's Level-1 LSP database is full and the BigIron RX receives an additional LSP, for which there is no room.
Level-2 Database Overload	The number of times the Level-2 state on the BigIron RX changed from Waiting to On or from On to Waiting. • The change from Waiting to On can occur when the BigIron RX 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 BigIron RX's Level-2 LSP database is full and the BigIron RX receives an additional LSP, for which there is no room.
Our LSP Purged	The number of times the BigIron RX received an LSP that was originated by the BigIron RX itself and had age zero (aged out).

Clearing IS-IS information

To clear the IS-IS information that the device 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 a command such as the following at any level of the CLI except the User EXEC level.

```
BigIron RX# clear isis all
```

This command clears all the following.

- Neighbors (closes the BigIron RX'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 BigIron RX's adjacencies with its IS-IS neighbors and clears the 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.

Chapter

BiDirectional Forwarding Detection (BFD) 30

In this chapter

BigIron RX provides support for Bidirectional Forwarding Detection (BFD) in Version 02.6.00 of the Multi-Service IronWare software. BFD defines a method of rapid detection of the failure of a forwarding path by checking that the next hop router is alive. Without BFD enabled, it can take from 3 to 30 seconds to detect that a neighboring router 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 router transmits control messages at a high rate of speed that is negotiated by the routers 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. The maximum number of sessions supported on an interface module (LP) is 20 while the maximum per system is 100.

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 trunks in cases where all BFD packet are transmitted only on a single path which does not change unless the trunk active membership changes. BFD is not be supported on multi-slot trunks where per-packet switching is used such that the path taken by the BFD packets will vary per packet.

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.

```
BigIron RX(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 router 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 router waits to receive a BFD message from its peer. The router 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.

NOTE
The **transit-time** and **receive-time** variables set with this command are the intervals desired by the local router. The actual values in use will be the negotiated values.

The <*number*> variable specifies the number of times in a single sequence that this router 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.

Number of BFD sessions supported

The device supports a maximum of 100 BFD sessions per system with a maximum number of 20 sessions per interface module. This number is inclusive of the fact that IS-IS and OSPF sessions on an interface module will include both transmit and receive sessions. Consequently, the 20 sessions per interface module actually corresponds to 40 sessions where each OSPF and IS-IS session consumes two sessions (one transmit and one receive).

Disabling BFD Syslog messages

Beginning with version 02.6.00 of the Multi-Service IronWare software, syslog messages are generated for BFD operation. These messages are described in "Brocade Syslog messages". Logging of these messages is enabled by default. To disable logging of BFD messages use the following command.

```
BigIron RX(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 Bidirectional Forwarding Detection information

You can display Bidirectional Forwarding Detection (BFD) information for the router you are logged-in to and for BFD configured neighbors as described in the following sections.

Displaying BFD information on a router

The following example illustrates the output from the **show bfd** command.

```
BigIron RX# show bfd
 BFD State: ENABLED Version: 1
 Current Registered Protocols: ospf    ospf6
 All Sessions: Current: 2 Maximum Allowed: 100 Maximum Exceeded Count: 0
 LP Sessions: Maximum Allowed on LP: 20  Maximum Exceeded Count for LPs: 0
   LP Sessions LP Sessions LP Sessions LP Sessions
   1  0        2  2        3  0        4  0
   5  0        6  0        7  0        8  0
   9  0        10 0        11 0        12 0
   13 0        14 0        15 0        16 0
BFD Enabled ports count: 2
 Port       MinTx       MinRx       Mult Sessions
 eth 2/1    100         100         3    2
```

Syntax: show bfd

This display shows the following information.

TABLE 139 Display of BFD information

This field...	Displays...
BFD State	Specifies if BFD is Enabled or Disabled on the router.
Version	Specifies the version of the BFD protocol operating on the router.
Current Registered Protocols	Specifies which protocols are registered to use BFD on the router. Possible values are ospf, ospf6, or isis_task
All Sessions	
Current:	The number of BFD sessions currently operating on the router.
Maximum Allowed	The maximum number of BFD sessions that are allowed on the router. The maximum number of sessions supported on a router is 100.
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 router.
LP Sessions	
Maximum Allowed on LP	The maximum number of BFD sessions that are allowed on an interface module. The maximum number of sessions supported on an interface module is 20.
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.

TABLE 139 Display of BFD information (Continued)

This field...	Displays...
Sessions	The number of BFD sessions currently operating on the specified Interface module.
BFD Enabled ports count	The number of ports on the router that have been enabled for BFD.
Port	The port that BFD is enabled on.
MinTx	The interval in milliseconds between which the router desires to send a BFD message from this port to its peer.
MinRx	The interval in milliseconds that this router desires to receive a BFD message from its peer on this port.
Mult	The number of times that the router will wait for the MinRx time on this port before it determines that its peer router is non-operational.
Sessions	The number of BFD sessions originating on this port.

Displaying BFD application information

The following example illustrates the output from the **show bfd application** command.

```
BigIron RX# show bfd application
 Registered Protocols Count: 3
     Protocol       Parameter
     ospf                 1
     ospf6                0
     isis_task            0
```

This display shows the following information.

TABLE 140 Display of BFD application information

This field...	Displays...
Parameter	The parameter value passed by the protocol during registration with BFD.

Displaying BFD neighbor information

The following example illustrates the output from the **show bfd neighbor** command.

```
BigIron RX# 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 141 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 router.
NeighborAddress	The IPv4 or IPv6 address of the remote peer.
State	The current state of the BFD session. Up - Up Down - 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 after which the session will transition to the down state if no message is received.
Interval	The negotiated interval at which the local router sends BFD messages to the remote peer.
RH	Heard from remote.

To display BFD Neighbor information in the detailed format use the following command.

```
BigIron RX# 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: ospf
  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 [IpAddress | IPv6Address]

This display shows the following information.

TABLE 142 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 142 Display of BFD neighbor detail information (Continued)

This field...	Displays...
Holddown	The interval after which the session will transition to the down state if no message is received.
Interval	The interval at which the local router 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 router in the last message sent.
Diag	Value of the “diagnostic” field in the BFD Control Message as used by the local router in the last message sent.
Demand	Value of the “demand” bit in the BFD Control Message as used by the local router in the last message sent.
Poll	Value of the “poll” bit in the BFD Control Message as used by the local router in the last message sent.
MinTxInterval	The interval in microseconds between which the router negotiates to send a BFD message from this local neighbor port to its peer.
MinRxInterval	The interval in microseconds that the neighbor router waits to receive a BFD message from its peer on this local port.
Multiplier	The number of times that the neighbor router will wait for the MinRxInterval time on this port before it determines that its peer router 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 microseconds between which the router negotiates to send a BFD message from the remote neighbor port to its peer.
MinRxInterval	The interval in microseconds that the neighbor router waits to receive a BFD message from its peer on this remote port.
Multiplier	The number of times that the remote neighbor router will wait for the MinRxInterval time on this port before it determines that its peer router 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 142 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.

```
BigIron RX# 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

Configuring BFD for OSPFv2

You can configure your device router 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 router's OSPFv2 enabled interfaces using the command shown in the following"

```
BigIron RX (config)# router ospf
BigIron RX(config-ospf-router)# bfd all-interfaces
```

Syntax: [no] bfd all-interfaces

While this command configures BFD for OSPFv2 on all of a router's OSPFv2 enabled interfaces, it is not required that it be configured if you use the **ip ospf bfd** command to configure specific interfaces. It can be used independently or together with that command.

Enabling or disabling BFD for OSPFv2 for a specific interface

You can selectively enable or disable BFD on any OSPFv2 interface as shown in the following.

```
BigIron RX# (config-if-e1000-3/1)# ip ospf bfd
```

Syntax: ip ospf bfd [disable]

The **disable** option disables BFD for OSPFv2 on the interface.

Configuring BFD for OSPFv3

You can configure your BigIron RX router 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 of a router's OSPFv3 enabled interfaces using the command shown in the following"

```
BigIron RX(config)# ipv6 router ospf
BigIron RX(config-ospf6-router)# bfd all-interfaces
```

Syntax: [no] bfd all-interfaces

While this command configures BFD for OSPFv3 on all of a router's OSPFv3 enabled interfaces, it is not required that it be configured if you use the **ipv6 ospf bfd** command to configure specific interfaces. It can be used independently or together with that 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.

```
BigIron RX#(config-if-e1000-3/1)# ipv6 ospf bfd
```

Syntax: ipv6 ospf bfd [disable]

The **disable** option disables BFD for OSPFv3 on the interface.

Configuring BFD for IS-IS

You can configure your BigIron RX router for BFD 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 of a router's IS-IS enabled interfaces using the command shown in the following"

```
BigIron RX(config)# router isis
BigIron RX(config-isis-router)# bfd all-interfaces
```

Syntax: [no] bfd all-interfaces

While this command configures BFD for IS-IS on all of a router's IS-IS enabled interfaces, it is not required that it be configured if you use the **isis bfd** command to configure specific interfaces. It can be used independently or together with that 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.

```
BigIron RX#(config-if-e1000-3/1)# isis bfd
```

Syntax: isis bfd [disable]

The **disable** option disables BFD for IS-IS on the interface.

Chapter

Configuring Multi-Device Port Authentication 31

In this chapter

How multi-device port authentication works

Multi-device port authentication is a way to configure a BigIron RX to forward or block traffic from a MAC address based on information received from a RADIUS server. Multi-device port authentication is supported in the device software release 02.2.01 and later.

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 device supports 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 device device. 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 device device.

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 device 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.

BigIron RX Series support multi-device port authentication on untagged ports only.

Supported RADIUS attributes

The device supports 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 2579
- 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 device a RADIUS Access-Accept message that allows the device 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 BigIron RX device, 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 BigIron RX device.

In addition to dynamic VLAN assignment, BigIron RX Series 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 device, 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/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 device 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 device.

To globally enable multi-device port authentication on the device, enter the following command.

```
BigIron RX(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. Brocade 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.

For example.

```
BigIron RX(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 BigIron RX, you must identify the server to the device. For example.

```
BigIron RX(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 BigIron RX 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

Specifying the format of the MAC addresses sent to the RADIUS server

When multi-device port authentication is configured, the device 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.

```
BigIron RX(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.

```
BigIron RX(config)# interface e 3/1
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config)# interface e 3/1
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config)# interface e 3/1
BigIron RX(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 device a RADIUS Access-Accept message that allows the device 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 device device, 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 928 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.

```
BigIron RX(config)# interface e 3/1
BigIron RX(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 device 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.

```
BigIron RX(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 to which VLAN a port is moved after its 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 BigIron RX device, 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.

```
BigIron RX(config)# interface e 3/1
BigIron RX(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 device to save the RADIUS-specified VLAN assignments to the device's running configuration file. To do this, enter the following command.

```
BigIron RX(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 device 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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config)# interface e 3/1
BigIron RX(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 device 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. (See 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.

```
BigIron RX(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.

```
BigIron RX(config)# interface e 3/1
BigIron RX(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 device 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 device 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 device 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.

```
BigIron RX(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.

```
BigIron RX# 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 143 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.

```
BigIron RX# 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 144 Output from the show auth-mac-address configuration command

This field...	Displays...
Feature enabled	Whether the multi-device port authentication feature is enabled on the BigIron RX device.
Number of Ports enabled	The number of ports on which the multi-device port authentication feature is enabled.

TABLE 144 Output from the show auth-mac-address configuration command (Continued)

This field...	Displays...
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.
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.
Protect Limit	This is not applicable to the BigIron RX, 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.:

```
BigIron RX# show auth-mac-address detail
Port 1/18
 Dynamic-Vlan Assignment          : Enabled
 RADIUS failure action            : Block Traffic
 Override-restrict-vlan           : Yes
 Port VLAN                        : 4094 (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 145 Output from the show authenticated-mac-address command

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.
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.

TABLE 145 Output from the **show authenticated-mac-address** command (Continued)

This field...	Displays...
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.

```
BigIron RX# 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 146 Output from the **show auth-mac-address** *<address>* **command**

This field...	Displays...
MAC/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.
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.

```
BigIron RX# 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.

```
BigIron RX# 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

32

In this chapter

Overview of MAC port security

You can configure the BigIron RX 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 device 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: either drops packets from the violating address (and allows packets from the secure addresses), or disables the port altogether 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.

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 address using local resources.

Besides the maximum of 64 local resources available to an interface, there are additional global resources. Depending on flash memory size, a device can have 1024, 2048, or 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

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 at once or on individual interfaces.

To enable the feature globally.

```
BigIron RX(config)# global-port-security
BigIron RX(config-port-security)# enable
```

To disable the feature on all interfaces at once.

```
BigIron RX(config)# global-port-security
BigIron RX(config-port-security)# no enable
```

To enable the feature on a specific interface.

```
BigIron RX(config)# int e 7/11
BigIron RX(config-if-e100-7/11)# port security
BigIron RX(config-port-security-e100-7/11)# enable
```

Syntax: global-port-security

Syntax: port-security

Syntax: [no] enable

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.

```
BigIron RX(config)# int e 7/11
BigIron RX(config-if-e100-7/11)# port security
BigIron RX(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 2048 or 4096, depending on flash memory size. Setting the parameter to 0 prevents any addresses from being learned. The default is 1.

Setting the port security age timer

By default, the learned MAC addresses stay secure indefinitely. You can optionally configure the device to age out secure MAC addresses after a specified amount of time.

To set the port security age timer to 10 minutes globally.

```
BigIron RX(config)# global-port-security
BigIron RX(config-port-security)# age 10
```

To set the port security age timer to 10 minutes on a specific interface.

```
BigIron RX(config)# int e 7/11
BigIron RX(config-if-e100-7/11)# port security
BigIron RX(config-port-security-e100-7/11)# age 10
```

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.

```
BigIron RX(config)# int e 7/11
BigIron RX(config-if-e100-7/11)# port security
BigIron RX(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. For example, to automatically save learned secure MAC addresses on the device every twenty minutes, enter the following commands.

```
BigIron RX(config)# global-port-security
BigIron RX(config-port-security)# autosave 20
```

Syntax: [no] autosave <*minutes*>

You can specify from 15 – 1440 minutes. By default, secure MAC addresses are not autosaved to the startup-config file.

Defining security violation actions

A MAC port 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 MAC port security violation occurs, an SNMP trap and Syslog message are generated. Also, you can configure the device to take any of the following actions when a MAC port security violation occurs:

- **Violation restrict** – This action shuts the port down after denying a certain number of violating MACs
- **Violation shutdown** – This action shuts the port down on the first violation

Violation restrict

The violation restrict action shuts the port down after denying a certain number of violating MAC addresses. To enable this command, enter the following command.

```
BigIron RX(config)# int e 7/11
BigIron RX(config-if-e100-7/11)# port security
BigIron RX(config-port-security-e100-7/11)# violation restrict
BigIron RX(config-port-security-e100-7/11)#restrict-max-deny 130
```

Syntax: violation restrict

Syntax: restrict-max-deny <*number*>

The **violation restrict** command enables the violation restrict action.

The **restrict-mac-deny** command specifies the number of MAC addresses that are to be denied before the device shuts the port down. Enter 1 – 1024. The default is 128. In the example above, the port will be shut down after 130 MAC addresses are denied.

Violation shutdown

This violation shutdown action shuts the port down on the first violation. To enable this action, enter the following command.

```
BigIron RX(config)# int e 7/11
BigIron RX(config-if-e100-7/11)#port security
BigIron RX(config-port-security-e100-7/11)# violation shutdown
```

Syntax: violation shutdown

Port shutdown time

When you enable either the violation restrict or violation shutdown action, you can specify how long the action lasts. For example, you can enter commands such as the following.

```
BigIron RX(config)# int e 7/11
BigIron RX(config-if-e100-7/11)#port security
BigIron RX(config-port-security-e100-7/11)# violation shutdown
BigIron RX(config-port-security-e100-7/11)#shutdown-time
```

Syntax: shutdown-time

Enter 0 – 1440 minutes, with 0 as the default. Specifying 0 shuts down the port permanently when a MAC port security violation occurs.

The shutdown time applies to both the violation restrict and violation shutdown actions.

Re-enabling a port

Once a port is permanently shut down, an administrator must re-enable the port by entering the following command.

```
BigIron RX(config)# int e 7/11
BigIron RX(config-if-e100-7/11)#enable
```

Syntax: enable

Port security MAC violation limit

You can specify how many packets the system can receive in a one-second interval from denied MAC address before the system shuts the port down. To enable this new mode, enter a command such as the following.

```
BigIron RX(config)#global-port-security
BigIron RX(config-port-security)#violation restrict 12
```

Syntax: violation restrict [#-denied-packets processed]

Enter 1 – 64000. There is 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.

```
BigIron RX(config)#global-port-security
BigIron RX(config-port-security)#violation restrict 12
BigIron RX(config-port-security)#deny-log-rate <7>
```

Syntax: deny-log-rate [<*#-logs*>]

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]
```

Transparent port flooding

When the transparent port flooding feature in enabled for a port, all MAC learning will be disabled for that port. This will result in all Layer 2 traffic to be flooded to all other ports within the VLAN. The Transparent port flooding feature is disabled by default. To enable Transparent port flooding, enter a command such as the following:

```
BigIron RX (config-if-e1000-15/8)mac-learn-disable
```

Syntax: [no] mac-learn-disable

Displaying transparent port flooding status

To display if the transparent port flooding feature has been enabled, issue the **show interface ethernet** <*interface*> or **show running-configuration** command.

Example

```
10.45.48.50(config-if-e1000-15/8)#mac-learn-disable
10.45.48.50(config-if-e1000-15/8)#show int e 15/8
GigabitEthernet15/8 is down, line protocol is down
Hardware is GigabitEthernet, address is 0004.1234.ffff (bia 0004.1234.ffff)
Configured speed 1Gbit, actual unknown, configured duplex fdx, actual unknown
Configured mdi mode AUTO, actual unknown
Member of L2 VLAN ID 1, port is untagged, port state is Disabled
STP configured to ON, Priority is level0, flow control enabled
Force-DSCP disabled
SA learning is 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
```

Displaying MAC port security information

You can display the following information about the MAC 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.

```
BigIron RX# show port security e 7/11
Port  Security  MacAddrs   Violation PortShutdn(minutes)       SecureMac  Learn
              Learnt/Max Total/Count/Type Status/Time/          Remain  AgeTime
----- --------- ---------- --------------------------------- ---------  -----
15/1  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/2  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/3  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/4  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/5  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/6  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/7  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/8  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/9  disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/10 disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
15/11 disabled  0/1        0/ 0/shutdown no/permanent          permanent  yes
```

Syntax: show port security <*module*> | <*portnum*>

This command displays the following information.

TABLE 147 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.
Violation PortShutdn (minutes) Total/Count/Type	The total number of violation that has occurred. The action to be undertaken when a security violation occurs, either “shutdown” or “restrict”. The number of seconds a port is shut down following a security violation.
SecureMac Remain	How many minutes the restrict or shutdown action will be in effect. "Permanent" means the port is permanently shut down.
Learn Age-Time	The amount of time, in minutes, MAC addresses learned on the port will remain secure.

Displaying the secure MAC addresses on the device

To list the secure MAC addresses configured on the device, enter the following command.

```
BigIron RX(config)# show port security mac
Port  Count  Secure-Addr(S)       Vlan    AgeLeft
----- -----  ------------------   ----    ---------
3/2   1      0003.0000.0001 (S)   1       permanent
3/2   2      0003.0000.0002 (S)   1       permanent
3/2   3      0003.0000.0003 (S)   1       permanent
3/2   4      0003.0000.0004 (S)   1       permanent
```

Syntax: show port security mac

This command displays the following information.

TABLE 148 Output from the show port security mac command

This field...	Displays...
Port	The slot and port number of the interface.
Count	The number of MAC addresses secured on this interface.
Secure-Src-Addr (S)	The secure MAC address. (S) means "secure".
VLAN	ID of VLAN to which the port is assigned.
Age-Left	The number of minutes the MAC address will remain secure.

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.

```
BigIron RX# 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 149 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/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.

```
BigIron RX# 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 150 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.

Displaying a list of MAC addresses

To display a list of MAC addresses that are secure, enter the following commands.

```
BigIron RX#show mac
Total active entries from all ports = 8
MAC Address      Port    Age       VLAN    Type
0003.0000.0001   3/2     Secure     1      secure(Allow)
0003.0000.0003   3/2     Secure     1      secure(Allow)
0004.0000.0002   5/1     0          1
0004.0000.0004   5/1     0          1
0004.0000.0001   5/1     0          1

BigIron RX#show mac all
Total active entries from all ports = 10
MAC Address      Port     Age       VLAN   Type
0003.0000.0001   3/2      Secure    1      secure(Allow)
0003.0000.0003   3/2      Secure    1      secure(Allow)
0003.0000.000a   3/2      Secure    1      secure(Deny)
0003.0000.000b   3/2      Secure    1      secure(Deny)
0004.0000.0002   5/1      0         1
0004.0000.0004   5/1      0         1
0004.0000.0001   5/1      0         1
```

Syntax: show mac [all]

Entering **show mac** displays MAC addresses, excluding those denied when violation restrict is enabled. The **show mac all** command displays all MAC address entries, including those denied when violation resrtict is enabled.

Chapter

Configuring 802.1x Port Security

33

In this chapter

Overview of 802.1x port security

The *BigIron RX* software release 02.2.01 and later supports the IEEE 802.1x standard for authenticating devices attached to LAN ports. Using 802.1x port security, you can configure a *BigIron RX* 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 *BigIron RX* 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

Brocade's 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.

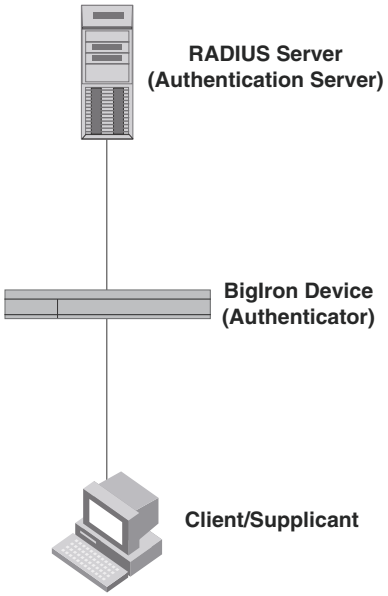
Device roles in an 802.1x configuration

The 802.1x standard defines the roles of ***Client/Supplicant***, ***Authenticator***, and ***Authentication Server*** in a network.

The Client (known as a ***Supplicant*** in the 802.1x standard) provides username/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 120 illustrates these roles.

FIGURE 120 Authenticator, Client/Supplicant, and Authentication Server in an 802.1x configuration



Authenticator – The device that controls access to the network. In an 802.1x configuration, the *BigIron RX* 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/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.

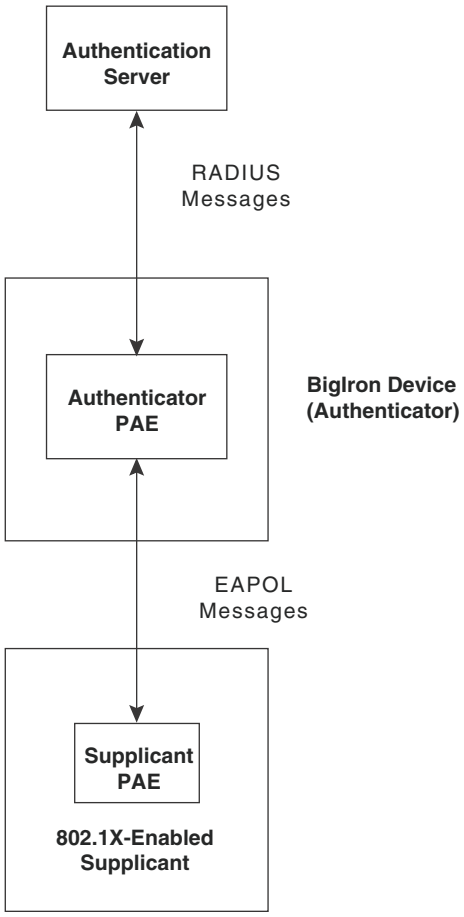
Authentication Server – The device that validates the Client and specifies whether or not the Client may access services on the device. *Brocade* supports Authentication Servers running RADIUS.

Communication between the devices

For communication between the devices, 802.1x port security uses the ***Extensible Authentication Protocol*** (EAP), 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/Supplicant, Authenticator, and Authentication Server.

EAPOL messages are passed between the ***Port Access Entity (PAE)*** on the Supplicant and the Authenticator. Figure 121 shows the relationship between the Authenticator PAE and the Supplicant PAE.

FIGURE 121 Authenticator PAE and supplicant PAE



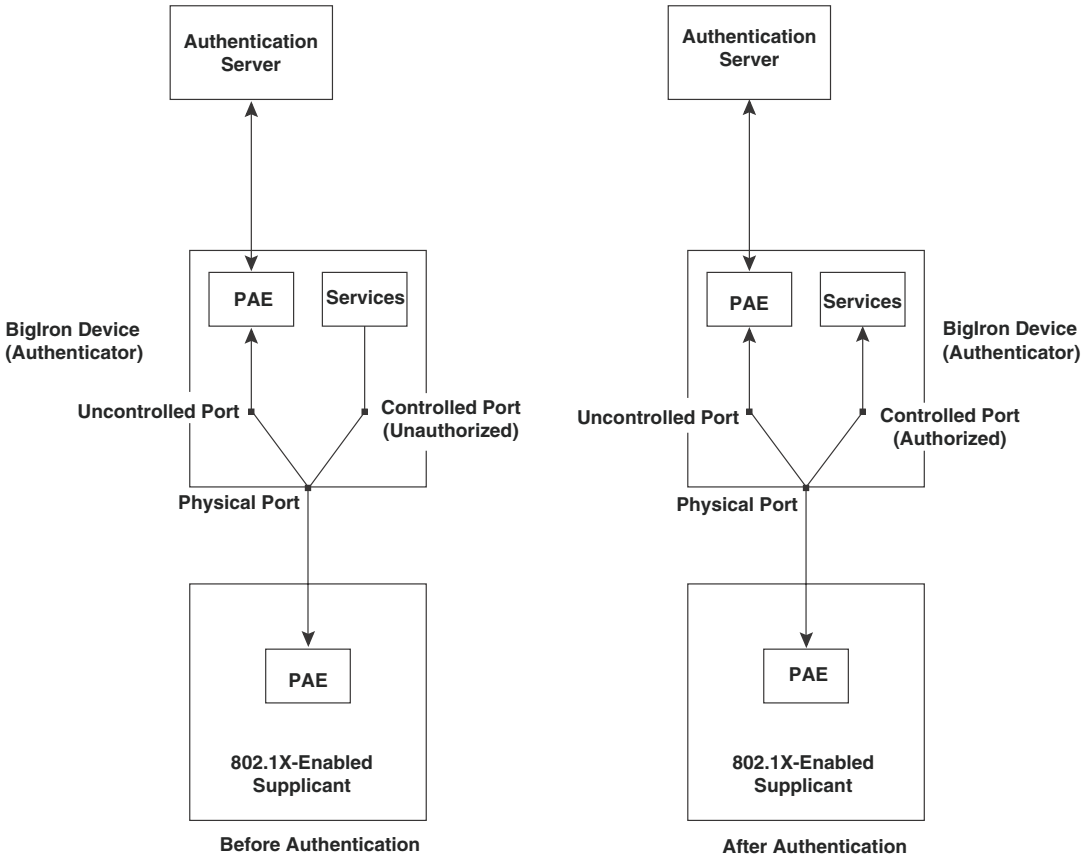
Authenticator PAE – The Authenticator PAE communicates with the Supplicant PAE, receiving identifying information from the Supplicant. Acting as a RADIUS client, the Authenticator PAE passes the Supplicant's information to the Authentication Server, which decides whether the Supplicant can gain access 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 122 illustrates this concept.

FIGURE 122 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.

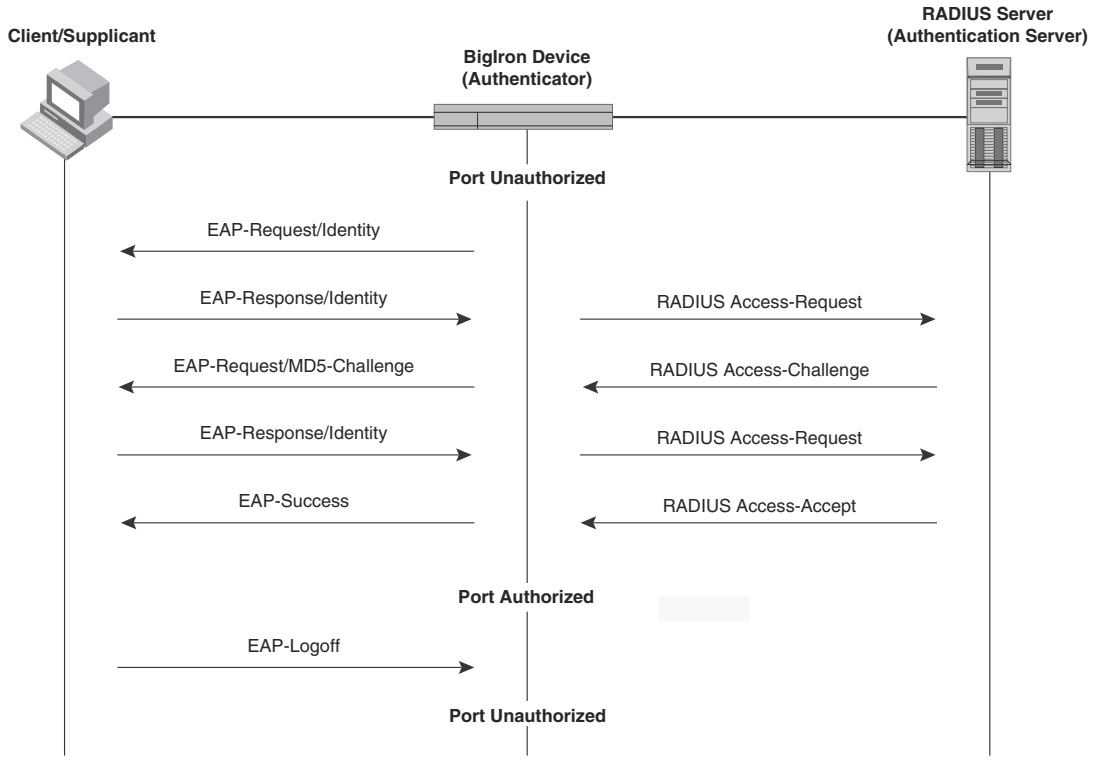
During authentication, EAPOL messages are exchanged between the Supplicant PAE and the Authenticator PAE, and RADIUS messages are exchanged between the Authenticator PAE and the Authentication Server. Refer to "Message exchange during authentication" on page 957 for an example of this process. If the 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 interface's controlled port 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 966 for more information.

Message exchange during authentication

Figure 123 illustrates a sample exchange of messages between an 802.1x-enabled Client, a *BigIron RX* acting as Authenticator, and a RADIUS server acting as an Authentication Server.

FIGURE 123 Message exchange between Client/Supplicant, Authenticator, and Authentication Server



In this example, the Authenticator (the *BigIron RX* 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.

Brocade'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 *BigIron RX* 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 962 for more information.

Brocade'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/Identity frames to the Client, but the Client does not respond to them.

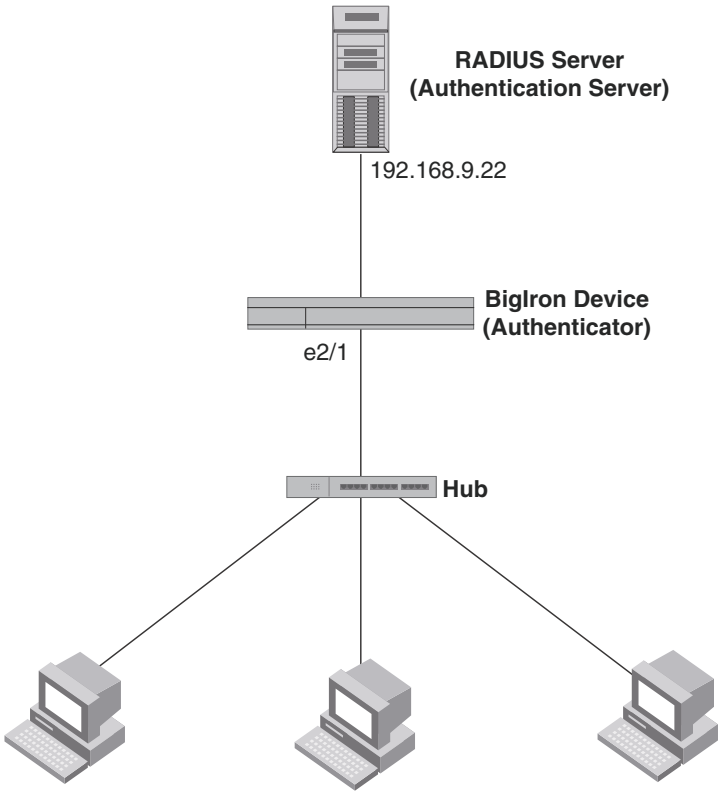
When a Client that supports 802.1x attempts to gain access through a non-802.1x-enabled port, it sends an EAP start frame to the *BigIron RX* device. When the device does not respond, the Client considers the port to be authorized, and starts sending normal traffic.

BigIron RX devices support MD5-challenge TLS and any other EAP-encapsulated authentication types in EAP Request/Response messages.In other words, the BigIron RX devices are transparent to the authentication scheme used.

Authenticating multiple clients connected to the same port

BigIron RX devices support 802.1x authentication for ports with more than one Client connected to them. Figure 124 illustrates a sample configuration where multiple Clients are connected to a single 802.1x port.

FIGURE 124 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 *BigIron RX* (as in Figure 124), 802.1x authentication is performed in the following way.

1. One of the 802.1x-enabled Clients attempts to log into a network in which a *BigIron RX* 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 971 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 971 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 *BigIron RX* 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.

Configuring 802.1x port security

Configuring 802.1x port security on a device consists of the following tasks.

1. Configuring the *BigIron RX* device's interaction with the Authentication Server:
   - "Configuring an authentication method list for 802.1x" on page 961
   - "Setting RADIUS parameters" on page 961
   - "Configuring dynamic VLAN assignment for 802.1x ports" on page 962 (optional)
2. Configuring the *BigIron RX*'s role as the Authenticator:
   - "Enabling 802.1x port security" on page 966
   - "Initializing 802.1x on a port" on page 970 (optional)
3. Configuring the *BigIron RX* device's interaction with Clients:
   - "Configuring periodic re-authentication" on page 968 (optional)
   - "Re-authenticating a port manually" on page 968 (optional)
   - "Setting the quiet period" on page 969 (optional)
   - "Setting the interval for retransmission of EAP-request/identity frames" on page 969 (optional)
   - "Specifying the number of EAP-request/identity frame retransmissions" on page 969 (optional)
   - "Specifying a timeout for retransmission of EAP-request frames to the client" on page 970 (optional)
   - "Allowing multiple 802.1x clients to authenticate" on page 970 (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/802.1x client.

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. *Brocade* 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 *BigIron RX* and RADIUS server.

For example.

```
BigIron RX(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 *BigIron RX*, you must identify the server to the device. For example.

```
BigIron RX(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 *BigIron RX* 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 *BigIron RX* 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

Brocade'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 *BigIron RX* 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/supplicant successfully completes the EAP authentication process, the Authentication Server (the RADIUS server) sends the Authenticator (the *BigIron RX*) 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 151 802.1x 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 *BigIron RX*.

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 975 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 *Brocade BigIron RX*, 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 *Brocade BigIron RX*, 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.

```
BigIron RX(config)# dot1x-enable
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(config)# interface e 1
BigIron RX(config-if-e10000-1)# no dot1x filter-strict-security
```

To re-enable strict security mode for an interface, enter the following command.

```
BigIron RX(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 *Brocade* IP ACL or MAC address filter.

The following is the syntax for configuring the Filter-ID attribute to refer to a *Brocade* 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.
mac.<*number*>.in	Applies the specified numbered 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 *BigIron RX*.

Possible values for the filter ID attribute on the RADIUS server	ACL or MAC address filter configured on the *BigIron RX* 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 eq 0800
mac.402.in mac.403.in	access-list 402 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800 access-list 403 permit 2222.2222.2222 ffff.ffff.ffff any etype eq 0800

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 only for the inbound direction. Dynamic outbound ACL filters are not supported.
- 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 filters to be simultaneously applied to an 802.1x authenticated port. Use commas, semicolons, or carriage returns to separate the filters (for example ip.3.in,mac.2.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 *Brocade* ACL or MAC address filter statements. When the RADIUS server returns the Access-Accept message granting a client access to the network, the *BigIron RX* 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 *BigIron RX* 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.
macfilter.in=<*mac-filter-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 *Brocade* Vendor-Specific attribute on a RADIUS server. These IP ACLs and MAC address filters follow the same syntax as other *Brocade* ACLs and MAC address filters. Refer to Chapter 21, "Access Control List" for information on syntax.

Mac address filter	Vendor-specific attribute on RADIUS server
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 *BigIron RX* devices. To enable the feature on the device and enter the dot1x configuration level, enter the following command.

```
BigIron RX(config)# dot1x-enable
BigIron RX(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.

```
BigIron RX(config-dot1x)# enable all
```

Syntax: [no] enable all

To enable 802.1x port security on interface 3/11, enter the following command.

```
BigIron RX(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.

```
BigIron RX(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, a controlled port and an uncontrolled port:

- 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 122 on page 956 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.

```
BigIron RX(config)# interface e 3/1
BigIron RX(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 *BigIron RX*. 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:

- 10 Gbps ports

- Static MAC configurations
- Link aggregation
- Metro Ring Protocol (MRP)
- Tagged port
- Mirror port
- Trunk 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.

```
BigIron RX(config)#dot1x-enable
BigIron RX(config-dot1x)# re-authentication
```

Syntax: [no] re-authentication

To configure periodic re-authentication with an interval of 2,000 seconds, enter the following commands.

```
BigIron RX(config)#dot1x-enable
BigIron RX(config-dot1x)# re-authentication
BigIron RX(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. See "Re-authenticating a port manually", below.

Re-authenticating a port manually

When periodic re-authentication is enabled, by default the *BigIron RX* 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.

```
BigIron RX# dot1x re-authenticate e 3/1
```

Syntax: [no] dot1x re-authenticate <*portnum*>

Setting the quiet period

If the *BigIron RX* 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.

```
BigIron RX(config-dot1x)# timeout quiet-period 30
```

Syntax: [no] timeout quiet-period <*seconds*>

Setting the interval for retransmission of EAP-request/identity frames

When the *BigIron RX* sends a Client an EAP-request/identity frame, it expects to receive an EAP-response/identity frame from the Client. If the Client does not send back an EAP-response/identity frame, the device waits a specified amount of time and then retransmits the EAP-request/identity frame. You can specify the amount of time the *BigIron RX* waits before retransmitting the EAP-request/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 *BigIron RX* to wait 60 seconds before retransmitting an EAP-request/identity frame to a Client, enter the following command.

```
BigIron RX(config-dot1x)# timeout tx-period 60
```

Syntax: [no] timeout tx-period <*seconds*>

If the Client does not send back an EAP-response/identity frame within 60 seconds, the device will transmit another EAP-request/identity frame.

Specifying the number of EAP-request/identity frame retransmissions

If the *BigIron RX* does not receive a EAP-response/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/identity frame. By default, the *BigIron RX* retransmits the EAP-request/identity frame a maximum of two times. If no EAP-response/identity frame is received from the Client after two EAP-request/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/identity frame to a Client a maximum of three times, enter the following command.

```
BigIron RX(config-dot1x)# maxreq 3
```

Syntax: maxreq <*value*>

Specifying a timeout for retransmission of messages to the authentication server

When performing authentication, the *BigIron RX* 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 *BigIron RX* 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 *BigIron RX*, 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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX# 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 *BigIron RX* 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 959 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.

```
BigIron RX(config)# dot1x-enable
BigIron RX(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.

```
BigIron RX(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 authentication-failure action is to drop traffic from the Client, and the initial authentication attempt made by the device to authenticate the Client is unsuccessful, the *BigIron RX* immediately retries to authenticate the Client. After three unsuccessful authentication attempts, the Client's dot1x-mac-session is set to "access-denied", causing traffic from the Client to be dropped in hardware.

You can optionally configure the number of authentication attempts the device makes before dropping traffic from the Client. To do so, enter a command such as the following.

```
BigIron RX(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 before dropping packets from the 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 mac** 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. For example.

```
BigIron RX# 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 *BigIron RX* device, enter the following command.

```
BigIron RX# 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 152 Output from the show dot1x command

This field...	Displays...
PAE Capability	The Port Access Entity (PAE) role for the *BigIron RX* 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.

TABLE 152 Output from the show dot1x command (Continued)

This field...	Displays...
re-authentication	Whether periodic re-authentication is enabled on the device. Refer to "Configuring periodic re-authentication" on page 968. 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 *BigIron RX* is unable to authenticate a Client, the amount of time the *BigIron RX* waits before trying again (default 60 seconds). Refer to "Setting the quiet period" on page 969 for information on how to change this setting.
tx-period	When a Client does not send back an EAP-response/identity frame, the amount of time the *BigIron RX* waits before retransmitting the EAP-request/identity frame to a Client (default 30 seconds). Refer to "Setting the interval for retransmission of EAP-request/identity frames" on page 969 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 *BigIron RX* retransmits the frame. Refer to "Specifying a timeout for retransmission of EAP-request frames to the client" on page 970 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 *BigIron RX* retransmits the message. Refer to "Specifying a timeout for retransmission of messages to the authentication server" on page 970 for information on how to change this setting.
max-req	The number of times the *BigIron RX* retransmits an EAP-request/identity frame if it does not receive an EAP-response/identity frame from a Client (default 2 times). Refer to "Specifying the number of EAP-request/identity frame retransmissions" on page 969 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 968 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.

```
BigIron RX# 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 153 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 *BigIron RX* through a hub. Refer to "Allowing multiple 802.1x clients to authenticate" on page 970 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.

```
BigIron RX# 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 154 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/Id	The number of EAP-Response/Identity frames received on the port
RX EAP Resp other than Resp/Id	The total number of EAPOL-Response frames received on the port that were not EAP-Response/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/Id	The number of EAP-Request/Identity frames transmitted on the port.
TX EAP Req other than Req/Id	The number of EAP-Request frames transmitted on the port that were not EAP-Request/Identity frames.
Num sessions	Total number of dot1x sessions, which include authenticated, restricted, denied and sessions in the initial state.
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.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX#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.

```
BigIron RX#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.

```
BigIron RX#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.

```
BigIron RX#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.

```
BigIron RX# 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 155 describes the information displayed by the **show dot1x mac-session** command.

TABLE 155 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.

```
BigIron RX# 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 156 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.

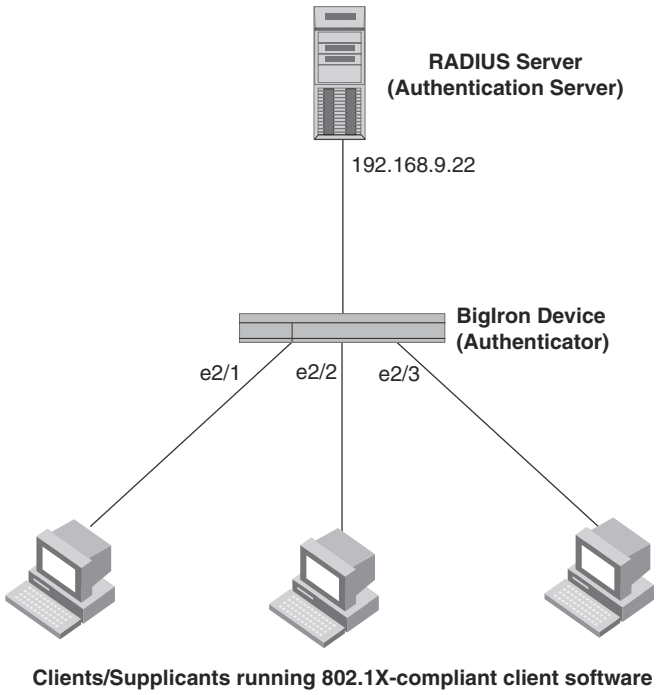
Sample 802.1x configurations

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 125 illustrates a sample 802.1x configuration with Clients connected to three ports on the *BigIron RX* device. In a point-to-point configuration, only one 802.1x Client can be connected to each port.

FIGURE 125 Sa*mple point-to-point 802.1x configura*tion



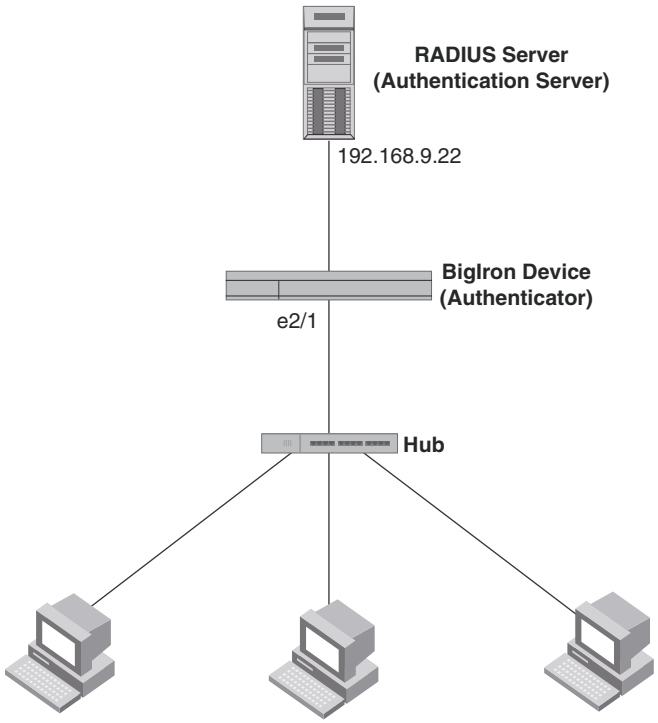
The following commands configure the *BigIron RX* in Figure 125.

```
BigIron RX(config)# aaa authentication dot1x default radius
BigIron RX(config)# radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x
BigIron RX(config)# dot1x-enable e 2/1 to 2/3
BigIron RX(config-dot1x)# re-authentication
BigIron RX(config-dot1x)# timeout re-authperiod 2000
BigIron RX(config-dot1x)# timeout quiet-period 30
BigIron RX(config-dot1x)# timeout tx-period 60
BigIron RX(config-dot1x)# max-req 6
BigIron RX(config-dot1x)# exit
BigIron RX(config)# interface e 2/1
BigIron RX(config-if-e100-1)# dot1x port-control auto
BigIron RX(config-if-e100-1)# exit
BigIron RX(config)# interface e 2/2
BigIron RXconfig-if-e100-2)# dot1x port-control auto
BigIron RX(config-if-e100-2)# exit
BigIron RX(config)# interface e 2/3
BigIron RX(config-if-e100-3)# dot1x port-control auto
BigIron RX(config-if-e100-3)# exit
```

Hub configuration

Figure 126 illustrates a configuration where three 802.1x-enabled Clients are connected to a hub, which is connected to a port on the *BigIron RX* device. The configuration is similar to that in Figure 125, 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 126 Sample 802.1x configuration using a hub



The following commands configure the *BigIron RX* in Figure 126.

```
BigIron RX(config)# aaa authentication dot1x default radius
BigIron RX(config)# radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x
BigIron RX(config)# dot1x-enable e 2/1
BigIron RX(config-dot1x)# re-authentication
BigIron RX(config-dot1x)# timeout re-authperiod 2000
BigIron RX(config-dot1x)# timeout quiet-period 30
BigIron RX(config-dot1x)# timeout tx-period 60
BigIron RX(config-dot1x)# max-req 6
BigIron RX(config-dot1x)# exit
BigIron RX(config)# interface e 2/1
BigIron RX(config-if-e100-1)# dot1x port-control auto
BigIron RX(config-if-e100-1)# exit
```

Chapter

Protecting Against Denial of Service Attacks 34

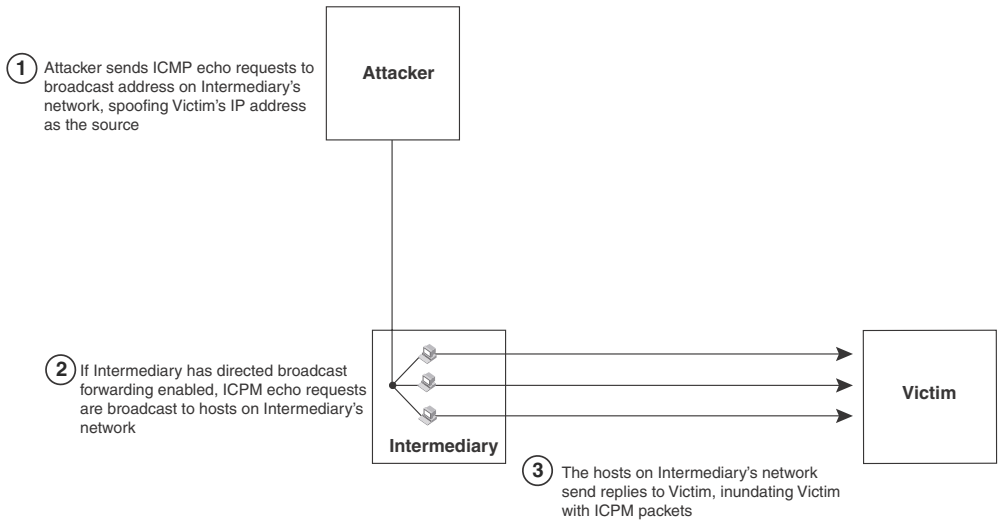
In this chapter

In a Denial of Service (DoS) attack, a router is flooded with useless packets, hindering normal operation. The *BigIron RX* includes measures for defending 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 (Ping) replies sent from another network. Figure 127 illustrates how a Smurf attack works.

FIGURE 127 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 *BigIron RX*. Directed broadcast forwarding is disabled by default. To disable directed broadcast forwarding, do the following.

```
BigIron RX(config)# no ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

ACL-based DOS-attack prevention

ACL-based DOS-attack prevention provides great flexibility on what packets can be rate-limited or dropped up. In fact, users can create any matching conditions they want to regulate any particular traffic flow they have in mind. This section provides examples that can be used to prevent two common types of DOS attacks.

Avoiding being a victim in a Smurf attack

You can configure the *BigIron RX* to drop ICMP packets when excessive numbers are encountered, as is the case when the device is the victim of a Smurf attack. You can set threshold values for ICMP packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

For example, to set threshold values for ICMP packets received on interface 3/11, enter the following command.

```
BigIron RX(config)# access-list 101 permit icmp any any echo-reply
BigIron RX(config)# int e 3/11
BigIron RX(config-if-e100-3/11)# dos-attack-prevent 101 burst-normal 5000000
burst-max 1000 lockup 300
```

In the example, if the total traffic volume of ICMP echo-reply packets received per second exceeds 5,000,000 bits per second, the excess packets are dropped. If the number of ICMP echo-reply packets received per second exceeds 1,000, the device drops all ICMP packets for the next 300 seconds (five minutes).

Syntax: dos-attack-prevent <*num*> burst-normal <*bps*> burst-max <*num-of-packets*> lockup <*seconds*> [log]

<*num*> is the ACL ID that will be used to check for traffic conformance.

The parameters **burst-normal**, **burst-max**, and **lockup** are applied individually on each ACL filter.

The **burst-normal** value, 1 – 100000000, is specified as bits per second.

The **burst-max** value, 1 – 100000, is specified as number of packets.

The **lockup** value can be from 1 – 10000 seconds.

The number of incoming ICMP packets that match the condition specified in the ACL per second are measured and compared to the threshold values as follows:

- If the total traffic volume (in bits per second) of packets that match the condition specified in the ACL exceeds the **burst-normal value**, the excess packets are dropped.
- If the number of packets that match the condition specified in the ACL exceeds the **burst-max** value, *all* packets that match the condition specified in the ACL 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.

When a port is locked up by dos-attack prevention, two types of syslog messages will be generated. The first type of messages will be generated at the time the port is shut down for the matched traffic flow to indicate the port shutdown activity and the period of shutdown. The following is a sample output.

```
Jun 23 00:40:20:N:Incoming traffic in interface 3/5 exceedes 1500 burst packets,
stopping for 30 seconds!!
```

The second type of messages will log the headers of the packets that are dropping during the lockup period. Note that this kind of messages are rate-limited to avoid overloading the syslog buffer. By default the same kind of packets will only be logged once every five seconds. The rate of the messages can be changed by the **ip access-list logging-age** command, which also controls the logging timer for ACL. The following is a sample output.

```
Jun 23 00:37:58:I:list 120 denied icmp 55.55.55.1()(Ethernet 3/5 0000.0000.0011)
-> 14.14.14.1(), 1 event(s)
```

Note that:

- This feature is supported on Ethernet(physical) interfaces only.
- Only the permit clauses (filters) are used in this feature. Deny clauses are ignored.

Protecting against TCP SYN attacks

TCP SYN attacks exploit the process of how TCP connections are established in order to disrupt normal traffic flow. When a TCP connection starts, the connecting host first 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 around a 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 the *BigIron RX* to drop TCP SYN packets when excessive numbers are encountered. You can set threshold values for TCP SYN packets that are targeted at the router itself or passing through an interface from interface 3/11, and drop them when the thresholds are exceeded.

For example, to set threshold values for TCP SYN packets, enter the following commands.

```
BigIron RX(config)# access-list 101 permit tcp any any match-all +syn
BigIron RX(config)# int e 3/11
BigIron RX(config-if-e100-3/11)# dos-attack-prevent 101 burst-normal 5000000
burst-max 1000 lockup 300
```

TCP security enhancement

TCP security enhancement improves upon the handling of TCP inbound segments. The 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, wherein 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. Also, the attacker 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 three 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 987.

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 in order 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 *BigIron RX* silently drops the segment.
- If the RST bit is exactly the next expected sequence number, the *BigIron RX* 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 *BigIron RX* sends an acknowledgement.

This TCP security enhancement is enabled by default. To disable it, refer to "Disabling the TCP security enhancement" on page 987.

Protecting against a blind TCP reset attack using the SYN bit

In a blind TCP reset attack, a perpetrator attempts to guess the SYN bits to prematurely terminate an active TCP session.

To prevent a user from using the SYN bit to tear down a TCP connection, the SYN bit is subject to the following rules when receiving TCP segments:

- If the SYN bit is set and the sequence number is outside the expected window, the device sends an acknowledgement (ACK) back 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 one from the value being acknowledged.
- If the SYN bit is set and the sequence number is acceptable, the device sends an acknowledgement (ACK) segment to the peer.

The TCP security enhancement is enabled by default. To disable it, refer to "Disabling the TCP security enhancement" on page 987.

Protecting against a blind injection attack

In a blind TCP injection attack, a perpetrator tries to inject or manipulate data in a TCP connection.

To reduce the chances of a blind injection attack, perform an additional check 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 987.

Disabling the TCP security enhancement

The TCP security enhancement is automatically enabled. If necessary, you can disable this feature. When you disable this feature, the *BigIron RX* reverts to the original behavior.

To disable the TCP security enhancement, enter the following command at the Global CONFIG level of the CLI.

```
BigIron RX(config)# no ip tcp tcp-security
```

To re-enable the TCP security enhancement once it has been enabled, enter the following command.

```
BigIron RX(config)# ip tcp tcp-security
```

Syntax: [no] ip tcp tcp-security

Displaying statistics due DoS attacks

To display information about ICMP and TCP SYN packets dropped, passed, and block because burst thresholds were exceeded.

```
BigIron RX(config-if-e1000-3/5)# show statistics dos-attack
Collecting transit DOS attack statistic for port 3/5...  Completed successfully.
------ DOS Attack Prevention Statistics ------
Port   Packet Drop Count    Packet Pass Count     Port Block Count
-----  -----------------    ------------------    ----------------
3/5              12479732              436372                 232
```

The display shows the following.

Port	Port number
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.
Port Block Count	Number of times the port was shut down for the particular traffic flow that matched the ACL.

Syntax: show statistics dos-attack [| begin <*expression*> | exclude <*expression*> | include <*expression*>]

Clear DoS attack statistics

To clear statistics about ICMP and TCP SYN packets dropped because burst thresholds were exceeded.

```
BigIron RX(config)# clear statistics dos-attack
```

Syntax: clear statistics dos-attack

Chapter

35

Inspecting and Tracking DHCP Packets

In this chapter

The features described in this chapter were introduced in software release 02.3.00 for the *BigIron RX* Series devices.

For enhanced network security, you can configure the *Brocade* device to inspect and keep track of Dynamic Host Configuration Protocol (DHCP) assignments. To do so, use the following features.

TABLE 157 Chapter contents

Description	See page
Dynamic ARP Inspection – Intercepts and examines all ARP request and response packets in a subnet, and blocks all packets that have invalid IP to MAC address bindings	page 989
DHCP Snooping – Filters replay DHCP packets from untrusted ports (those connected to host ports), and allows DHCP packets from trusted ports (those connected to DHCP servers)	page 994
IP Source Guard – Permits traffic with valid source IP addresses only, which is learned from Dynamic ARP Inspection or DHCP snooping	page 999

Dynamic ARP inspection

NOTE
This feature is only supported on Layer 3 code.

Dynamic ARP Inspection (DAI) enables the *Brocade* 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 attacks

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 ARP table, it sends 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 ARP 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 *Brocade* device on which DAI is configured does the following:

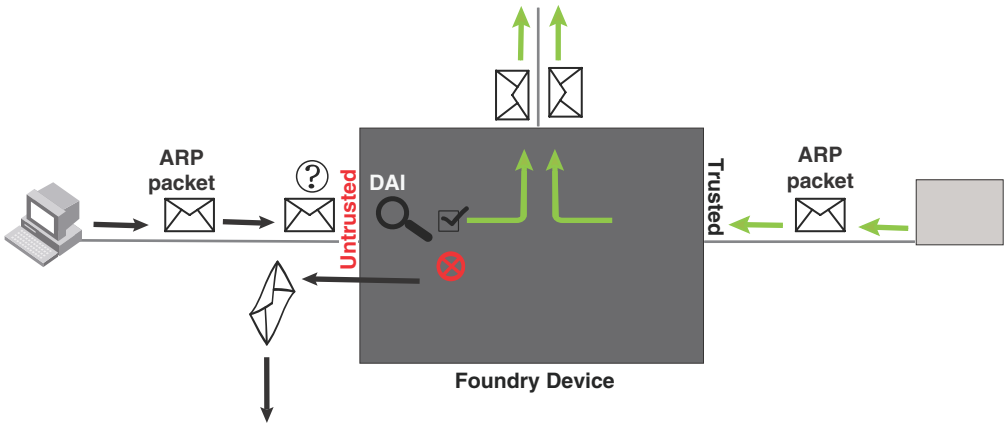
- Intercepts ARP packets received by the system CPU.
- Inspects all ARP requests and responses received on untrusted ports.
- Verifies that each of the intercepted packets has a valid IP-to-MAC address binding before updating the local ARP table, or before forwarding the packet to the appropriate destination.
- Drops invalid ARP packets

When you enable DAI 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 128. DAI carries out the inspection based on IP-to-MAC address bindings stored in a trusted binding database. For the *BigIron RX*, 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 128.

FIGURE 128 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:

- **Dynamic ARP** – normal ARP learned from trusted ports.
- **Static ARP** – statically configured IP/MAC/port mapping.
- **Inspection ARP** – statically configured IP/MAC mapping, where the port is initially unspecified. The actual physical port mapping will be resolved and updated from validated ARP packets. Refer to "Configuring an inspection ARP entry" on page 992.
- **DHCP-Snooping ARP** – information collected from snooping DHCP packets when DHCP snooping is enabled on VLANs.

The status of an ARP entry is either pending or valid:

- **Valid** – the mapping is valid, and the port is resolved. This is always the case for static ARP entries.
- **Pending** – for normal dynamic, inspection ARP, and DHCP-Snooping ARP entries before they are resolved, and the port mapped. Their status changes to valid when they are resolved, and the port mapped.

Refer to "System reboot and the binding database" on page 995.

Limits and restrictions

The following limits and restrictions apply when configuring DAI:

- The maximum number of DHCP and static DAI entries depends on the maximum number of ARP table entries allowed on the device. The BigIron RX Series switch can have up to 64,000 ARP entries. In a BigIron RX, you can use the **system-max ip-arp** command to change the maximum number of ARP entries for the device.

Configuring DAI

Configuring DAI consists of the following steps.

1. Configure inspection ARP entries for hosts on untrusted ports. Refer to "Configuring an inspection ARP entry" on page 992.
2. Enable DAI on a VLAN to inspect ARP packets. Refer to "Enabling DAI on a VLAN" on page 992.
3. 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. Refer to "Enabling trust on a port" on page 992.
4. Enable DHCP snooping to populate the DHCP snooping IP-to-MAC binding database.

The following shows the default settings of DAI.

Feature	Default
Dynamic ARP Inspection	Disabled
Trust setting for ports	Untrusted

Configuring an inspection ARP entry

Static ARP and static inspection ARP entries need to be configured for hosts on untrusted ports. Otherwise, when DAI checks ARP packets from these hosts against entries in the ARP table, it will not find any entries for them, and the *Brocade* device will not allow and learn ARP from an untrusted host.

When the inspection ARP entry is resolved with the correct IP/MAC mapping, its status changes from pending to valid.

To configure an inspection ARP entry, enter commands such as the following.

```
BigIron RX(config)#arp 20.20.20.12  0001.0002.0003 inspection
```

The commands defines an inspection ARP entry, mapping a device's IP address 20.20.20.12 with its MAC address 0001.0002.0003.

Syntax: [no] arp <*index*> <*ip-addr*> <*mac-addr*> inspection

The index can be from 1 up to the maximum number of static entries allowed.

The <*ip-addr*> <*mac-addr*> parameter specifies a device's IP address and MAC address pairing.

Enabling DAI on a VLAN

DAI is disabled by default. To enable DAI on an existing VLAN, enter the following command.

```
BigIron RX(config)#ip arp inspection vlan 2
```

The command enables DAI on VLAN 2. ARP packets from untrusted ports in VLAN 2 will undergo DAI inspection.

Syntax: [no] ip arp inspection vlan <*vlan-number*>

The <*vlan-number*> variable specifies the ID of a configured VLAN.

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.

```
BigIron RX(config)#interface ethernet 1/4
BigIron RX(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

Displaying ARP inspection status and ports

To display the ARP inspection status for a VLAN and the trusted or untrusted ports in the VLAN, enter the following command.

```
FastIron SuperX Switch#show ip arp inspection vlan 2
IP ARP inspection VLAN 2: Disabled
  Trusted Ports :   ethe 1/4
  Untrusted Ports : ethe 2/1 to 2/3 ethe 4/1 to 4/24 ethe 6/1 to 6/4 ethe 8/1 to
 8/4
```

Syntax: show ip arp inspection [vlan <*vlan_id*>]

The <*vlan_id*> variable specifies the ID of a configured VLAN.

Displaying the ARP table

To display the ARP table, enter the following command.

```
BigIron RX#show arp
Total number of ARP entries: 10
      IP Address          MAC Address      Type     Age      Port       Status
1     20.20.20.39         0000.4623.ed91   Dhcp     3/3386   5/20     Valid
2     20.20.20.40         0000.4623.ed8f   Dhcp     3/3386   5/20     Valid
3     20.20.20.41         0000.4623.ed8d   Dhcp     6/3321   5/20     Valid
4     20.20.20.42         0000.4623.ed8b   Inspect  4        5/20     Valid
5     20.20.20.43         0000.2698.7027   Inspect  4        5/20     Valid
6     20.20.20.44         0000.4623.ed89   Inspect  63       5/20     Pending
7     20.20.20.45         0000.4623.ed87   Inspect  182      5/20     Pending
8     10.43.11.1          0004.80a0.4000   Dynamic  0        mgmt1    Valid
9     10.43.11.45         0060.e040.a0c4   Dynamic  1        mgmt1    Valid
10    60.60.60.209        00d0.09a0.bd84   Inspect  14       0/64     Pending
```

The command displays all ARP entries in the system.

Syntax: show arp

TABLE 158 show arp command

This field...	Displays....
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Age	The ARP Age, which can be one of the following: • The number of minutes the entry has remained unused. If this value reaches the ARP aging period of 10 minutes, the entry is removed from the table. • The Inspect Pending entries are never removed from the ARP Table and are displayed in seconds not minutes. • The Inspect Valid entries are displayed in minutes and after 10 minutes of aging may be changed from Valid to Pending. • The DHCP age is in the form of x/y where x represents the ARP age in minutes and y represents the lease time remaining of the client. **NOTE:** Static entries do not age out.
Port	This field shows the port on which the entry was learned. .

TABLE 158 show arp command (Continued)

This field...	Displays....
Type	The ARP type, which can be one of the following: • Dynamic – The Layer 3 Switch learned the entry from an incoming packet on a trusted port. • Inspect (Inspection ARP) – The entry from a statically configured IP/MAC mapping, where the port was initially unspecified. • Dhcp (DHCP-Snooping ARP) – The Layer 3 Switch learned the entry from DHCP.
Status	The status, which can be one of the following: • Valid – The ARP entry was resolved with the correct IP/MAC mapping. Static ARP entries are always valid. • Pending – The ARP entry is not yet resolved.

DHCP snooping

NOTE
This feature is only supported on Layer 3 code.

Dynamic Host Configuration Protocol (DHCP) snooping enables the *Brocade* 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.

Often DHCP snooping is used together with Dynamic ARP Inspection and IP Source Guard.

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 129 DHCP snooping at Work - on untrusted port

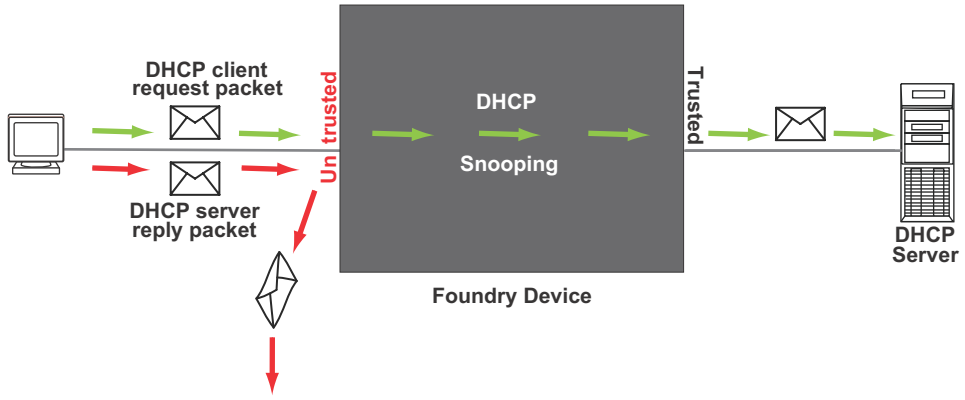
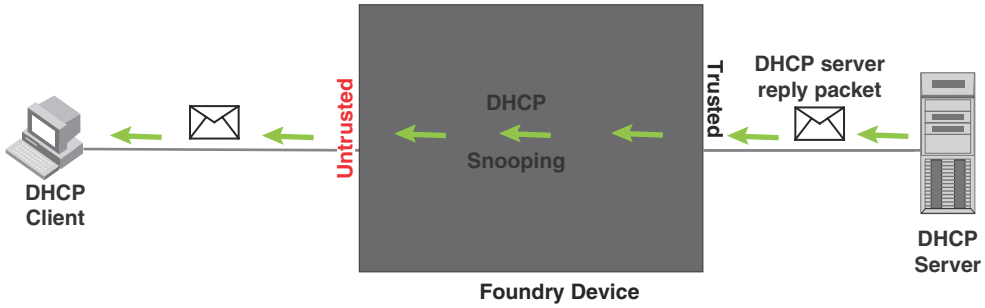


FIGURE 130 DHCP snooping at Work - on trusted port



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 learnt 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

Configuring DHCP snooping consists of the following steps.

1. Enable DHCP snooping on a VLAN. Refer to "Enabling DHCP snooping on a VLAN" on page 996.
2. For ports that are connected to a DHCP server, change their trust setting to trusted. Refer to "Enabling trust on a port" on page 996.

The following 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.

```
FastIron SuperX Switch(config)#ip dhcp snooping vlan 2
```

The command enables DHCP snooping on VLAN 2.

Syntax: [no] ip dhcp snooping vlan <*vlan-number*>

The <*vlan-number*> variable specifies the ID of a configured client or DHCP server VLAN.

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.

```
FastIron SuperX Switch(config)#interface ethernet 1/1
FastIron SuperX Switch(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

DHCP relay agent information (DHCP option 82)

DHCP relay agent information (DHCP option 82) 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.

This *Brocade* device inserts DHCP option 82 when relaying DHCP request packets to DHCP servers, and deletes option 82 when forwarding server reply packets back to DHCP clients. See the following figures.

FIGURE 131 DHCP option 82 is added to the packet

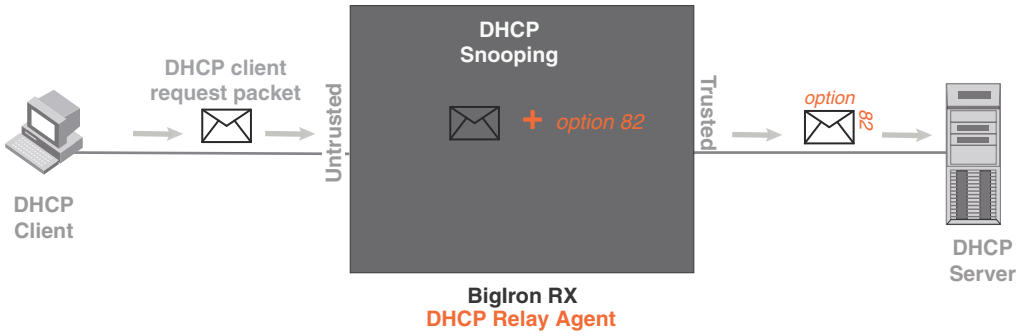
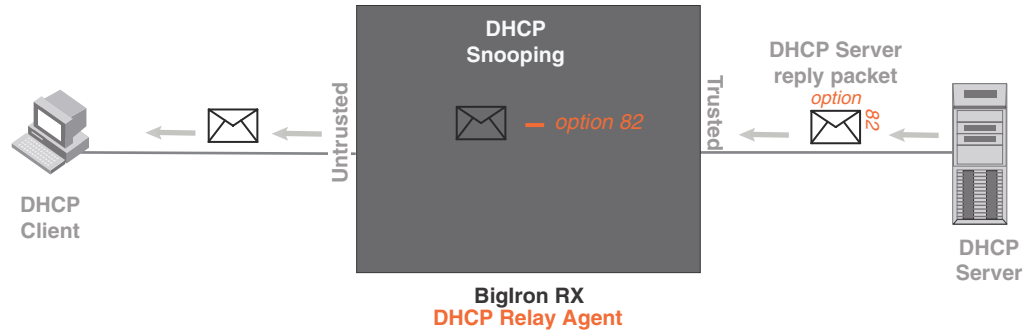


FIGURE 132 DHCP Option 82 Is Removed from the Packet



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.

DHCP option 82 contains sub-options such as the following:

- Supports sub-option 1, the relay agent circuit ID, with the sub-option data in the following format.

  VLAN id (2 bytes) / module id (1 byte) / port id (1 byte)

  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.

Disabling option 82 processing

When DHCP snooping is enabled on the *Brocade* device, option 82 processing is enabled by default.

To disable option 82 processing, enter the following commands.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e10000-1/1)# no dhcp relay information
```

The commands change the CLI to the interface configuration level for port 1/1 and disable option 82 processing on port 1/1.

Syntax: [no] dhcp relay information

Displaying DHCP snooping status and ports

To display the DHCP snooping status for a VLAN and the trusted or untrusted ports in the VLAN, enter the following command.

```
BigIron RX#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*>]

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.

```
FastIron SuperX Switch(config)#vlan 2
FastIron SuperX Switch(config-vlan-2)#untagged ethe 1/3 to 1/4
FastIron SuperX Switch(config-vlan-2)#router-interface ve 2
FastIron SuperX Switch(config-vlan-2)#exit
FastIron SuperX Switch(config)# ip dhcp snooping vlan 2

FastIron SuperX Switch(config)#vlan 20
FastIron SuperX Switch(config-vlan-20)#untagged ethe 1/1 to 1/2
FastIron SuperX Switch(config-vlan-20)#router-interface ve 20
FastIron SuperX Switch(config-vlan-20)#exit
FastIron SuperX Switch(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.

```
FastIron SuperX Switch(config)#interface ethernet 1/1
FastIron SuperX Switch(config-if-e1000-1/1)#dhcp snooping trust
FastIron SuperX Switch(config-if-e1000-1/1)#exit
FastIron SuperX Switch(config)#interface ethernet 1/2
FastIron SuperX Switch(config-if-e1000-1/2)#dhcp snooping trust
FastIron SuperX Switch(config-if-e1000-1/2)#exit
```

Hence, DHCP sever reply packets received on ports 1/1 and 1/2 are forwarded, and client IP/MAC binding information is collected.

The example also sets the DHCP server address for the local relay agent.

```
FastIron SuperX Switch(config)# interface ve 2
FastIron SuperX Switch(config-vif-2)#ip address 20.20.20.1/24
FastIron SuperX Switch(config-vif-2)#ip helper-address 30.30.30.4
FastIron SuperX Switch(config-vif-2)#interface ve 20
FastIron SuperX Switch(config-vif-20)#ip address 30.30.30.1/24
```

IP source guard

You can use IP Source Guard together with Dynamic ARP Inspection on untrusted ports. Refer to "DHCP snooping" on page 994 and "Dynamic ARP inspection" on page 989.

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.

When IP source guard is first enabled, the client port allows only DHCP packets, and blocks all other IP traffic. When the system learns a valid IP address on the port, the client port then allows IP traffic. Client ports permit only the traffic with valid source IP addresses.

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 991) When it learns a valid IP address, the system loads a per-port IP ACL entry permitting the learned source IP address on the port.

When a new IP source entry binding on the port is created or deleted, the per-port IP ACL will be recalculated and reapplied in hardware to reflect the change in IP source binding.

By default, if the IP source guard is enabled without any IP source binding on the port, an ACL that denies all IP traffic is loaded on the port. Similarly, when the IP source guard is disabled, any IP source per-port IP ACL will be removed from the interface.

Limits and restrictions

Current implementation with this feature has the following limitations:

- Works only on routing and virtual interface ports, and does not support Layer 2 switching-only ports in VLANs without an assigned IP address on the router.
- Does not support auto-saving of the learnt ARP entries when DAI is enabled. You must manually save the ARP entries before a reboot.
- Does not provide CLI to disable check for source MAC and source IP in DAI.

Enabling IP source guard

DHCP Snooping should be configured before you enable the IP source guard feature.

The default setting is disabled. To enable IP source guard on an untrusted port, enter the following commands.

```
BigIron RX(config)# interface ethernet 1/4
BigIron RX(config-if-e10000-1/4)# source guard enable
```

The commands change the CLI to the interface configuration level for port 1/4 and enable IP source guard on the port.

Syntax: [no] source guard enable

Displaying learned IP addresses

To display all IP source bindings configured on all interfaces on a switch, enter a command such as the following.

```
BigIron RX#show ip source guard eth 5/20
IP source guard on ethernet 5/20: Enabled
```

Syntax: show ip source guard ethernet <*port-num*>

Chapter

Securing SNMP Access 36

In this chapter

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. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters.

This chapter introduces a few methods used to secure SNMP access to the *BigIron RX*.

Establishing SNMP community strings

SNMP versions 1 and 2 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”. To open an SNMP session, enter “get” and “public” for the user name and password.
- By default, you cannot open a read-write management session. 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.

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

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.

Encryption is enabled by default. You can disable encryption for individual strings or trap receivers if desired. See the next section for information about encryption.

Adding an SNMP community string

When you add a community string, you can specify whether the string is encrypted or clear. By default, the string is encrypted.

To add an encrypted community string, enter commands such as the following.

```
BigIron RX(config)# snmp-server community private rw
BigIron RX(config)# write memory
```

The commands add the read-write SNMP community string “private” and saves it.

Syntax: snmp-server community [0] <*string*>
ro | rw [view <*viewname*>] [<*standard-acl-name*> | <*standard-acl-id*>]

By default, the community string is encrypted. When you save the new community string to the startup configuration file, the software adds the following command to the file.

```
snmp-server community 1 <encrypted-string> rw
```

If you want to create a non-encrypted community string, use the **0** option. as in the following example.

```
BigIron RX(config)# snmp-server community 0 private rw
BigIron RX(config)# write memory
```

The command in the example above adds the string “private” in the clear, which means the string is displayed in the clear text form. When you save the community string to the starup configuration file, the software adds the following command to the file.

```
snmp-server community 0 private rw
```

The <*string*> parameter specifies the community string name. The string can be up to 32 characters long.

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.

```
BigIron RX(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 “Defining SNMP views” on page 1009.

The <*standard-acl-name*> | <*standard-acl-id* parameter is optional. It allows you to specify which ACL will be used to filter incoming SNMP packets. You can enter either the ACL name or its ID. Here are examples.

```
BigIron RX(config) # snmp-s community myread ro view sysview 2
BigIron RX(config) # snmp-s community myread ro view sysview myacl
```

The command in the first example indicates that ACL group 2 will filter incoming SNMP packets, whereas the command in the second example uses the ACL group called “myacl” to filter incoming packets. Refer to “Using ACLs to restrict SNMP access” on page 65 for more information.

Displaying the SNMP community strings

To display the configured community strings, enter the following command at any CLI level.

```
BigIron RX(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 "Defining SNMP views" on page 1009.)

NOTE
SNMP version 3 Notification is not supported at this time. The system will generate traps in SNMP version 1 format.

NOTE
SNMP may timeout when trying to get module temperature values. You must increase the timeout value to 10 seconds to prevent a timeout.

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 *BigIron RX*.

Configuring SNMP version 3 on the *BigIron RX*

To configure SNMP version 3 on the *BigIron RX*, do the following.

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 1004.
2. Create views that will be assigned to SNMP user groups using the **snmp-server view** command. Refer to "Defining SNMP views" on page 1009 for details.
3. Create ACL groups that will be assigned to SNMP user groups using the **access-list** command. Refer to Chapter 21, "Access Control List" for details.
4. Create user groups using the **snmp-server group** command. Refer to "Defining an SNMP group" on page 1005.
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 1006.

If SNMP version 3 is not configured, then community strings by default are used to authenticate access.

Defining the engine ID

A default engine ID is generated during system start up. 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 "Displaying the engine ID" on page 1007 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.

```
BigIron RX(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. There are two hexadecimal characters in each octet. There should be an even number of hexadecimal characters in an engine ID.

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.

```
BigIron RX(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*>]

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 1001.) 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 or not authentication will be required to access 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 will be 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 <*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 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 allows SNMP version 3 to be backwards compatibility with SNMP version 1 and version 2.

NOTE
If you will be using a view other than the "all" view, that view must be configured before creating the user group. Refer to "Defining SNMP views" on page 1009, especially 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.
- Specifies one of the following encryption types used to encrypt the privacy password:
  - Data Encryption Standard (DES) – A symmetric-key algorithm that uses a 56-bit key.
  - Advanced Encryption Standard (AES) – The 128-bit encryption standard adopted by the U.S. government. This standard is a symmetric cipher algorithm chosen by the National Institute of Standards and Technology (NIST) as the replacement for DES.

Here is an example of how to create the account.

```
BigIron RX(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*> | 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, then the ACL configured for the group will be 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 will convert the password string to a digest, as described in RFC 3414.

The **auth md5 | sha** parameter is optional. It defines the type of encryption that the user must have to be authenticated. Choose between MD5 or SHA encryption. MD5 and SHA are 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] des <*des-password*> parameter is optional after you enter the md5 or sha password. The priv parameter defines the type of encryption that will be used to encrypt the privacy password. If the "encryption" keyword is used, enter a 16-octet DES key in hexadecimal format for the des-password. If the "encryption" keyword is not used, enter a password string of at least 8 characters. The agent will generate a suitable 16-octet DES key from the password string:

- If DES is the privacy protocol to be used, enter **des** followed by 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** followed by the AES password key. For a small password key, enter 12 characters. For a big password key, enter 16 characters. 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.

```
BigIron RX(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.

```
BigIron RX(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.

Displaying user information

To display the definition of an SNMP user account, enter a command such as the following.

```
BigIron RX(config)# show snmp user
username = bob
acl id = 2
group = admin
security model = v3
group acl id = 0
authtype = md5
authkey = 3aca18d90b8d172760e2dd2e8f59b7fe
privtype = des,  privkey = 1088359afb3701730173a6332d406eec
engine ID= 800007c70300e052ab0000
```

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
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 may also be generated: • If the configured ACL for this user filters out this packet. • If the 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 *BigIron RX*. This number cannot be changed.

To create an SNMP view, enter one of the following commands:

```
BigIron RX(config)# snmp-server view Maynes system included
BigIron RX(config)# snmp-server view Maynes system.2 excluded
BigIron RX(config)# snmp-server view Maynes 2.3.*.6 included
BigIron RX(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 *Brocade* MIBs objects that begin with the 1.3.6.1.4.1.1991 object identifier. Enter the following command.

BigIron RX`(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.

BigIron RX`(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

```
BigIron RX(config)#snmp-server group admingrp v3 priv read all write all notify
all
BigIron RX(config)#snmp-server user adminuser admingrp v3 auth md5 admin priv
admin1
BigIron RX(config)#snmp-server host 10.3.1.44
```

More detailed SNMP v3 configuration

```
BigIron RX(config)#snmp-server view internet internet included
BigIron RX(config)#snmp-server view system system included
BigIron RX(config)#snmp-server community ..... ro
BigIron RX(config)#snmp-server community ..... rw
BigIron RX(config)#snmp-server contact isc-operations
BigIron RX(config)#snmp-server location sdh-pillbox
BigIron RX(config)#snmp-server host 128.91.255.32 .....
BigIron RX(config)#snmp-server group ops v3 priv read internet write system
BigIron RX(config)#snmp-server group admin v3 priv read internet write internet
BigIron RX(config)#snmp-server group restricted v3 priv read internet
BigIron RX(config)#snmp-server user ops ops v3 encrypted auth md5
ab8e9cd6d46e7a270b8c9549d92a069 priv encrypted des
0e1b153303b6188089411447dbc32de
BigIron RX(config)#snmp-server user admin admin v3 encrypted auth md5
0d8a2123f91bfbd8695fef16a6f4207b priv encrypted des
18e0cf359fce4fcd60df19c2b6515448
BigIron RX(config)#snmp-server user restricted restricted v3 encrypted auth md5
261fd8f56a3ad51c8bcec1e4609f54dc priv encrypted des
d32e66152f89de9b2e0cb17a65595f43k
```

Chapter

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

In this chapter

This chapter discusses the Discovery Protocol (FDP) – a protocol used by *Brocade* devices to advertise themselves to other *Brocade* devices, and Cisco Discovery Protocol (CDP) – a protocol used by Cisco devices to advertise themselves to other Cisco devices. *Brocade* devices use this protocol to learn device and interface information for Cisco devices in the network.

Using FDP

FDP enables *Brocade* devices to advertise themselves to other *Brocade* devices on the network. When you enable FDP on a *Brocade* 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 *Brocade* device running FDP sends FDP updates on Layer 2 to MAC address 01-E0-52-CC-CC-CC. Other *Brocade* 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 *BigIron RX* 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.

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 *Brocade* device to globally send FDP packets, enter the following command at the global CONFIG level of the CLI.

```
BigIron RX(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 commands such as the following.

```
BigIron RX(config)# int e 2/1
BigIron RX(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 *BigIron RX* 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.

```
BigIron RX(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 *BigIron RX* 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.

```
BigIron RX(config)# fdp holdtime 360
```

Syntax: [no] fdp holdtime <*secs*>

The <*secs*> parameter specifies the number of seconds a *BigIron RX* 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 *Brocade* neighbors
- Individual FDP entries
- FDP information for an interface on the device you are managing
- FDP packet statistics

NOTE
If the *BigIron RX* has intercepted CDP updates, then the CDP information is also displayed.

Displaying neighbor information

To display a summary list of all the *Brocade* neighbors that have sent FDP updates to this *BigIron RX*, enter the following command.

```
BigIron RXA# 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
   -------------- ------------ ------ ---------- ----------- -------------
   BigIron RXB    Eth 2/9      178    Router     BigIron RX 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 159 Summary FDP and CDP neighbor information

This line...	Displays...
Device ID	The hostname of the neighbor.
Local Int	The interface on which this *BigIron RX* 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.

```
BigIron RXA# show fdp neighbor detail
Device ID: BigIronB configured as default VLAN1, tag-type8100
Entry address(es):
Platform: BigIron RX 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, Inc. 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 160 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 *BigIron RX* 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.

```
BigIron RXA# show fdp entry BigIron RXB
Device ID: BigIron RXB configured as default VLAN1, tag-type8100
Entry address(es):
Platform: BigIron RX 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, Inc. 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 160 on page 1014.

Displaying FDP information for an interface

To display FDP information for an interface, enter a command such as the following.

```
BigIron RXA# 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.

```
BigIron RXA# 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.

```
BigIron RX# 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.

```
BigIron RX# 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 *BigIron RX* forwards these packets without examining their contents. You can configure a *BigIron RX* 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.

BigIron RX supports intercepting and interpreting CDP version 1 and 2 packets.

NOTE
The *Brocade* 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 *BigIron RX* drops the packets. As a result, Cisco devices will no longer receive the packets.

Enabling interception of CDP packets globally

To enable the *BigIron RX* to intercept and display CDP packets, enter the following command at the global CONFIG level of the CLI.

```
BigIron RX(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 commands such as the following.

```
BigIron RX(config)# int e 2/1
BigIron RX(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 *BigIron RX* has learned from CDP packets, enter the following command.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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*>

To display CDP entries for a specific device, specify the device ID. Here is an example.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# clear fdp table
```

Syntax: clear fdp table

To clear CDP statistics, enter the following command.

```
BigIron RX# clear fdp counters
```

Syntax: clear fdp counters

Chapter

Remote Network Monitoring 38

In this chapter

Basic management

This chapter describes the remote monitoring features available on *Brocade* products.

The following sections contain procedures for basic system management tasks.

Viewing system information

You can access software and hardware specifics for a *BigIron RX*.

To view the software and hardware details for the system, enter the **show version** command.

```
BigIron RX# 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 *BigIron RX* and by configuration level.

To determine the available show commands for the system or a specific level of the CLI, enter the following command.

```
BigIron RX# 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 *BigIron RX*. 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.

```
BigIron RX# 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 *Brocade* 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 *BigIron RX*.

No configuration is required to activate collection of statistics for the *BigIron RX*. This activity is by default automatically activated at system start-up.

You can view a textual summary of the statistics for all ports by entering the following CLI command.

```
BigIron RX(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:

- If the product is a *Stackable device*, the ports are numbered sequentially starting with 1.
- If the product is a *Chassis device*, 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 161 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.
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.

TABLE 161 Export configuration and statistics (Continued)

This line...	Displays...
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. **NOTE:** Not supported in BigIron RX
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 **NOTE:** Not supported in BigIron RX
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 **NOTE:** Not supported in BigIron RX.
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. **NOTE:** Not supported in BigIron RX.

TABLE 161 Export configuration and statistics (Continued)

This line...	Displays...
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. **NOTE:** Not supported in BigIron RX.
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:** Not supported in BigIron RX.

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 *BigIron RX* 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.

```
BigIron RX(config)# rmon history 1 interface 1 buckets 10 interval 10 owner nyc02
```

Syntax: rrmon 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.

```
BigIron RX(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 <t*hreshold-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.

```
BigIron RX(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

sFlow 39

sFlow is a system for observing traffic flow patterns and quantities within and among a set of *BigIron RX* devices. 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.

Configuration considerations

Consider the following when you configure sFlow:

- Sample data is collected from inbound traffic on ports enabled for sFlow. However, both traffic directions are counted for byte and packet counter statistics sent to the collector.
- sFlow packets are not forwarded on the management interface, you will need to configure the IP interface on a line module to forward the sFlow packets.
- sFlow can not be configured on Link Aggregation ports until Link Aggregation is established. Once Link Aggregation is established, then the sFlow parameter appears on the interface mode which is configured Link Aggregation.

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
The device uses the router ID only if the device also has an IP interface with the same address.

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 1029 and “Displaying sFlow information” on page 1034.

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.

NOTE
sFlow does not export packets through the management port.

NOTE
sFlow does not use the management IP as the agent IP.

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. *BigIron RX* 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.

Inbound port monitoring

Inbound port monitoring and sFlow are enabled on a per-port basis, but inbound port monitoring and sFlow cannot be enabled on the same port. If you enable port monitoring on a port, sFlow cannot be enabled on that port. A module can have some port sflow-enabled and some port inbound monitoring enabled.

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

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.

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.

To specify sFlow collectors, enter a command such as the following.

```
BigIron RX(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 1025.

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 *BigIron RX* 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 *BigIron RX* 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 *BigIron RX* 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.

```
BigIron RX(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 rate, 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
Brocade recommends 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.

```
BigIron RX(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 *BigIron RX*, 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.

```
BigIron RX(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".

Enabling sFlow forwarding

sFlow exports data only for the interfaces on which you enable sFlow forwarding. You can enable sFlow forwarding on the Ethernet 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 1025 for the source address requirements.

NOTE
When you enable sFlow forwarding on an 802.1x-enabled interface, the samples taken from the interface
include the username used to obtain access to the inbound or outbound ports, if that information is available. For information about 802.1x, refer to Chapter 33, "Configuring 802.1x Port Security".

Enabling sFlow forwarding

To enable sFlow forwarding, enter commands such as the following.

```
BigIron RX(config)# sflow enable
BigIron RX(config)# interface ethernet 1/1 to 1/8
BigIron RX(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

ACL-based inbound sFlow

NOTE
This feature is available only for IPv4.

Beginning with release 02.5.00b, the Multi-Service IronWare software supports using an IPv4 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, *Brocade* created the Proprietary Tag Type 1991 that encapsulates the sFlow samples obtained through ACLbased sFlow and separates them from the sequence flow of other sFlow samples. Figure 1 shows the format of an sFlow packet, which illustrates the differences between a standard sFlow payload and an ACL-based payload.

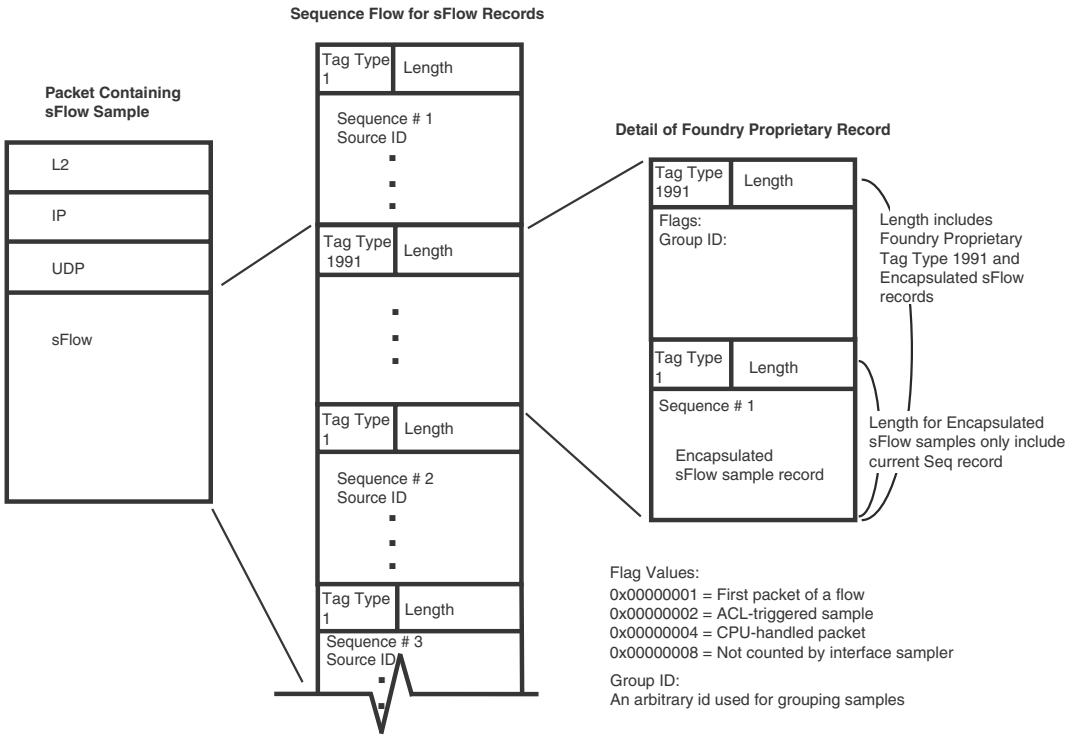
As shown in Figure 1, sFlow is carried in a UDP packet. Within the UDP packet, the sFlow contents are carried in individual samples that are identified by a Tag Type and a Length variable. The standard values for the Tag Types are 1 = sampled packet and 2 = counter sample. The length variable describes the length of the sample. Within the sample are other variables including the Sequence number and the Source ID.

Brocade 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 1, 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 that you might want to perform other processes.

FIGURE 133 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

Consider the following when you configure 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.

- **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 frowarding to the sFlow collector.
- **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.
- The sampling rate is the average ratio of the number of packets incoming on an sFlowenabled port, to the number of flow samples taken from those packets. However, for ACL based sFlow, every matching packet goes to the CPU. Consequently, configured sampling rates do not affect ACL based sFlow.

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.

```
BigIron RX(config)#access-list 151 permit tcp host 10.10.10.1 any established syn
copy-sflow
BigIron RX(config)#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.

```
BigIron RX(config)# int eth 1/1
BigIron RX(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.

To specify sFlow collectors, enter a command such as the following.

```
BigIron RX(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.

```
BigIron RX(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 162 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 1025.
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 *BigIron RX* has sent. **NOTE:** Each UDP packet can contain multiple samples.
sFlow samples collected	The number of sampled packets that have been sent to the collectors.
sFlow ports	The ports on which you enabled sFlow.

TABLE 162 sFlow information (Continued)

This field...	Displays...
Global Sample Rate	The global sampling rate for the *BigIron RX*.
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 sFlow information

To display sFlow configuration information and statistics, enter the following command at any level of the CLI.

```
BigIron RX(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 163 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 1025.
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 *BigIron RX* has sent. **NOTE:** Each UDP packet can contain multiple samples.
sFlow samples collected	The number of sampled packets that have been sent to the collectors.
sFlow ports	The ports on which you enabled sFlow.

TABLE 163 sFlow information (Continued)

This field...	Displays...
Global Sample Rate	The global sampling rate for the *BigIron RX*.
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

Clearing sFlow statistics

To clear the UDP packet and sFlow sample counters in the **show sflow** display, enter the following command.

```
BigIron RX(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.

39 sFlow

Chapter

Multiple Spanning Tree Protocol (MSTP) 802.1s 40

In this chapter

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.

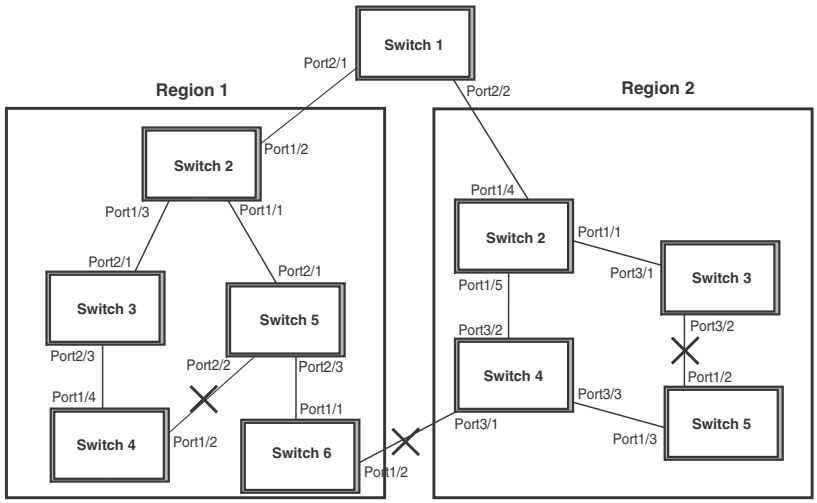
Multiple spanning-tree regions

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 134 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 134 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 4090. 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.

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.

```
BigIron RX(config)# mstp name foundry
```

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.

```
BigIron RX(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.

```
BigIron RX(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 **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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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.

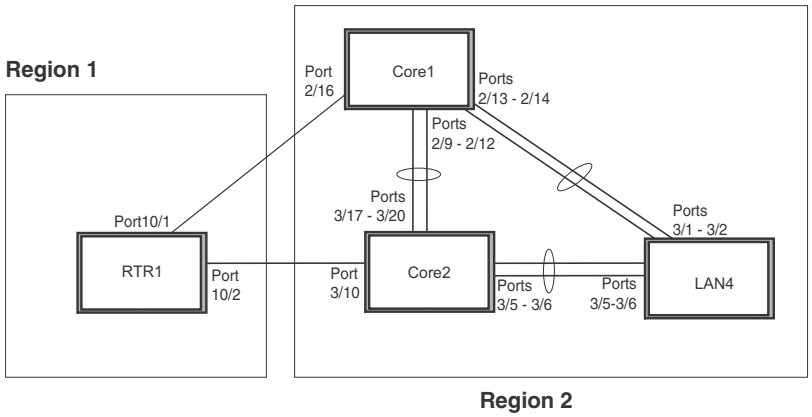
```
BigIron RX(config)#mstp start
```

Syntax: [no] start

Example

In Figure 135 four *BigIron RX*s are configured in two regions. There are four VLANs in four instances in Region 2. Region 1 is in the CIST.

FIGURE 135 SAMPLE MSTP configuration



RTR1 configuration

```
BigIron RX(config-vlan-4089)tagged ethe 10/1 to 10/2
BigIron RX(config-vlan-4089)exit
BigIron RX(config)mstp name Reg1
BigIron RX(config)mstp revision 1
BigIron RX(config)mstp instance 0 vlan 4089
BigIron RX(config)mstp admin-pt2pt-mac ethernet 10/1 to 10/2
BigIron RX(config)mstp start
BigIron RX(config)hostname RTR1
```

Core 1 configuration

```
BigIron RX(config)trunk switch ethernet 2/9 to 2/12 ethernet 2/13 to 2/14
BigIron RX(config)vlan 1 name DEFAULT-VLAN by port
BigIron RX(config-vlan-1)no spanning-tree
BigIron RX(config-vlan-1) exit
BigIron RX(config)vlan 20 by port
BigIron RX(config-vlan-20)tagged ethernet 2/9 to 2/14 ethernet 2/16
BigIron RX(config-vlan-20)no spanning-tree
BigIron RX(config-vlan-20) exit
BigIron RX(config)vlan 21 by port
BigIron RX(config-vlan-21)tagged ethernet 2/9 to 2/14 ethernet 2/16
BigIron RX(config-vlan-21)no spanning-tree
BigIron RX(config-vlan-21)exit
BigIron RX(config)vlan 22 by port
BigIron RX(config-vlan-22)tagged ethernet 2/9 to 2/14 ethernet 2/16
BigIron RX(config-vlan-22)no spanning-tree
BigIron RX(config-vlan-22)exit
BigIron RX(config)vlan 23 by port
BigIron RX(config) mstp name HR
BigIron RX(config) mstp revision 2
BigIron RX(config) mstp instance 20 vlan 20
BigIron RX(config) mstp instance 21 vlan 21
BigIron RX(config) mstp instance 22 vlan 22
BigIron RX(config) mstp instance 0 priority 8192
BigIron RX(config) mstp admin-pt2pt-mac ethernet 2/9 to 2/14
```

```
BigIron RX(config) mstp admin-pt2pt-mac ethernet 2/16
BigIron RX(config) mstp disable ethe 2/240
BigIron RX(config) mstp start
BigIron RX(config) hostname CORE1
```

Core2 configuration

```
BigIron RX(config)trunk switch ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
BigIron RX(config) vlan 1 name DEFAULT-VLAN by port
BigIron RX(config-vlan-1)no spanning-tree
BigIron RX(config-vlan-1) exit
BigIron RX(config)vlan 20 by port
BigIron RX(config-vlan-20)tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
BigIron RX(config-vlan-20)no spanning-tree
BigIron RX(config-vlan-20)exit
BigIron RX(config)vlan 21 by port
BigIron RX(config-vlan-21)tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
BigIron RX(config-vlan-21)no spanning-tree
BigIron RX(config-vlan-21)exit
BigIron RX(config) vlan 22 by port
BigIron RX(config-vlan-22)tagged ethe 3/5 to 3/6 ethe 3/17 to 3/20
BigIron RX(config-vlan-22)no spanning-tree
BigIron RX(config-vlan-22)exit
BigIron RX(config)mstp name HR
BigIron RX(config)mstp revision 2
BigIron RX(config)mstp instance 20 vlan 20
BigIron RX(config)mstp instance 21 vlan 21
BigIron RX(config)mstp instance 22 vlan 22
BigIron RX(config)mstp admin-pt2pt-mac ethernet 3/17 to 3/20 ethernet 3/5 to 3/6
BigIron RX(config)mstp admin-pt2pt-mac ethernet 3/10
BigIron RX(config)mstp disable ethe 3/7 ethernet 3/24
BigIron RX(config)mstp start
BigIron RX(config)hostname CORE2
```

LAN 4 configuration

```
BigIron RX(config) trunk switch ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
BigIron RX(config)vlan 1 name DEFAULT-VLAN by port
BigIron RX(config-vlan-1)no spanning-tree
BigIron RX(config-vlan-1)exit
BigIron RX(config)vlan 20 by port
BigIron RX(config-vlan-20)tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
BigIron RX(config-vlan-20)no spanning-tree
BigIron RX(config)exit
BigIron RX(config)vlan 21 by port
BigIron RX(config-vlan-21)tagged ethernet 3/1 to 3/2 ethe 3/5 to 3/6
BigIron RX(config-vlan-21)no spanning-tree
BigIron RX(config-vlan-21)exit
BigIron RX(config)vlan 22 by port
BigIron RX(config-vlan-22)tagged ethernet 3/1 to 3/2 ethe 3/5 to 3/6
BigIron RX(config-vlan-22)no spanning-tree
BigIron RX(config)mstp config name HR
BigIron RX(config)mstp revision 2
BigIron RX(config)mstp instance 20 vlan 20
BigIron RX(config)mstp instance 21 vlan 21
BigIron RX(config)mstp instance 22 vlan 22
BigIron RX(config)mstp admin-pt2pt-mac ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
BigIron RX(config)mstp start
BigIron RX(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.

```
BigIron RX(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 164 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.
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.

TABLE 164 Output from Show MSTP (Continued)

This field...	Displays...
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 802.1s 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.

```
BigIron RX(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 164 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.

```
BigIron RX(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.

```
BigIron RX(config)#show mstp conf
MSTP CONFIGURATION
------------------
Name     : Reg1
Revision : 1
Version  : 3 (MSTP mode)
Status   : Started

Instance VLANs
-------- ----------------------------------------------------
0        4089
```

To display details about the MSTP that is configured on the device, enter the following command.

```
BigIron RX(config)#show mstp detail
MSTP Instance 0 (CIST) - VLANs: 4089
--------------------------------------------------------------------------------
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 164 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*>.

Chapter

Configuring IP Multicast Traffic Reduction 41

In this chapter

The *BigIron RX* forwards all IP multicast traffic by default based on the Layer 2 information in the packets. Optionally, you can enable the device to make forwarding decisions in hardware, based on multicast group by enabling the IP Multicast Traffic Reduction feature.

When this feature is enabled, the device examines the MAC address in an IP multicast packet and forward the packet only on the ports from which the device has received Group Membership reports for that group, instead of forwarding all multicast traffic to all ports. The device sends traffic for other groups out all ports.

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.
- **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 *BigIron RX*.

NOTE
IP multicast traffic reduction and PIM SM Traffic Snooping is available on the *BigIron RX*.

Enabling IP multicast traffic reduction

By default, the *BigIron RX* forwards all IP multicast traffic out all ports except the port on which the traffic was received. To reduce multicast traffic through the device, you can enable IP 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 IGMP 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 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, the device 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.

NOTE
When one or more *BigIron RX* 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 *BigIron RX* devices for passive IGMP and allow the router to actively send the IGMP queries.

To enable IP Multicast Traffic Reduction, enter the following command.

```
BigIron RX(config)# ip multicast acitive
```

Syntax: [no] ip multicast active | passive

When you enable IP multicast on a *BigIron RX*, all ports on the device are configured for IGMP.

If you are using active IGMP, all ports can send IGMP queries and receive IGMP reports. If you are using passive IGMP, all ports can receive IGMP queries.

IP Multicast Traffic Reduction cannot be disabled on individual ports of a *BigIron RX*. IP Multicast Traffic Reduction can be disabled globally by entering the **no ip multicast** command.

To disable IP Multicast Traffic Reduction and IGMP snooping, enter the following command.

```
BigIron RX(config)# no ip multicast active
```

Syntax: [no] ip multicast

NOTE
If the "route-only" feature is enabled on the *BigIron RX*, then IP Multicast Traffic Reduction will not be supported.

Also, this feature is not supported on the default VLAN of the *BigIron RX*.

To verify that IP Multicast Traffic Reduction is enabled, enter the following command at any level of the CLI.

```
BigIron RX(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 IGMP modes are as follows:

- **Active** – When active IGMP mode is enabled, a *Brocade* 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 Switch*ed 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.

```
BigIron RX(config)# ip multicast active
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# reload
```

Syntax: [no] ip multicast active | passive

To enable passive IGMP, enter the following command.

```
BigIron RX(config)# ip multicast passive
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# reload
```

Configuring the IGMP mode per VLAN instance

If the IP Multicast command is not applied globally as described in "Enabling IP multicast traffic reduction" on page 1050, you can apply it to individual VLANs instances within their configurations. In the following example, multicast traffic reduction is applied using IGMP snooping to VLAN 2.

```
BigIron RX(config)# vlan 2
BigIron RX(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.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# no multicast
```

Syntax: [no] multicast active | passive

When you enable IP multicast for a specific VLAN 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 instance. You can use active or passive IGMP mode. There is no default mode.The IGMP modes are as follows:

- **Active** – When active IGMP mode is enabled, the router actively sends out IGMP queries to identify IP multicast groups within the VLAN 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 instance specified but does not send IGMP queries. The passive mode is called "IGMP snooping". Use this mode when another device in the VLAN instance is actively sending queries.

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 *BigIron RX* 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.

```
BigIron RX(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.

```
BigIron RX(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 *BigIron RX* forwards multicast traffic for all valid multicast groups. You can configure a *BigIron RX* 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.

```
BigIron RX(config)# ip multicast filter
```

Syntax: [no] ip multicast filter

Configuring IGMP snooping tracking per VLAN instance

When IGMP Snooping Tracking is enabled, the *BigIron RX* 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.

```
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX(config-vlan-100)# multicast tracking
```

Syntax: [no] multicast tracking

The **no** form of this command disables the tracking process per VLAN.

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

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 can be configured under the VLAN configuration only.

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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX((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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

```
BigIron RX(config)# vlan 100
BigIron RX(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.

PIM SM traffic snooping

By default, when a *BigIron RX* 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 *BigIron RX* 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.

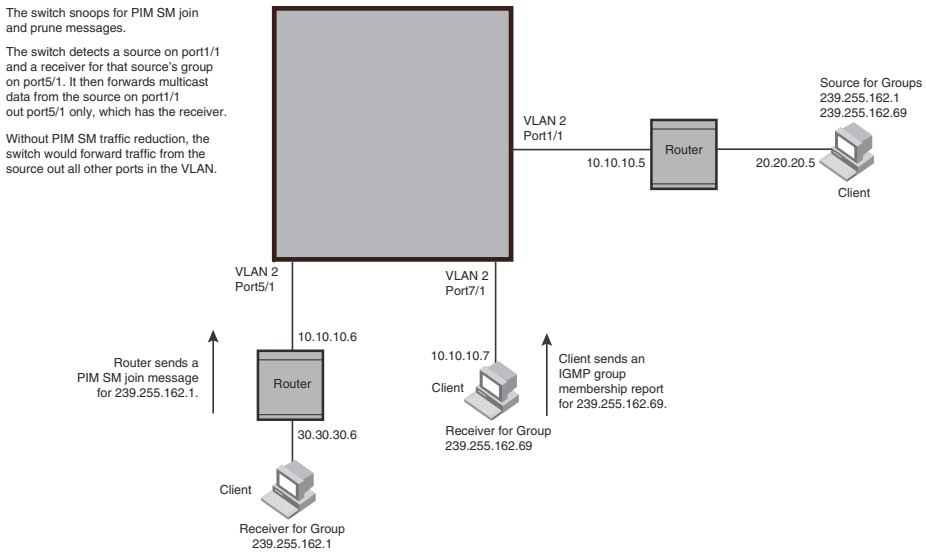
PIM SM traffic snooping requires IP multicast traffic reduction to be enabled on the device. IP multicast traffic reduction configures the device to listen for IGMP messages. PIM SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for 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 136 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 136 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.

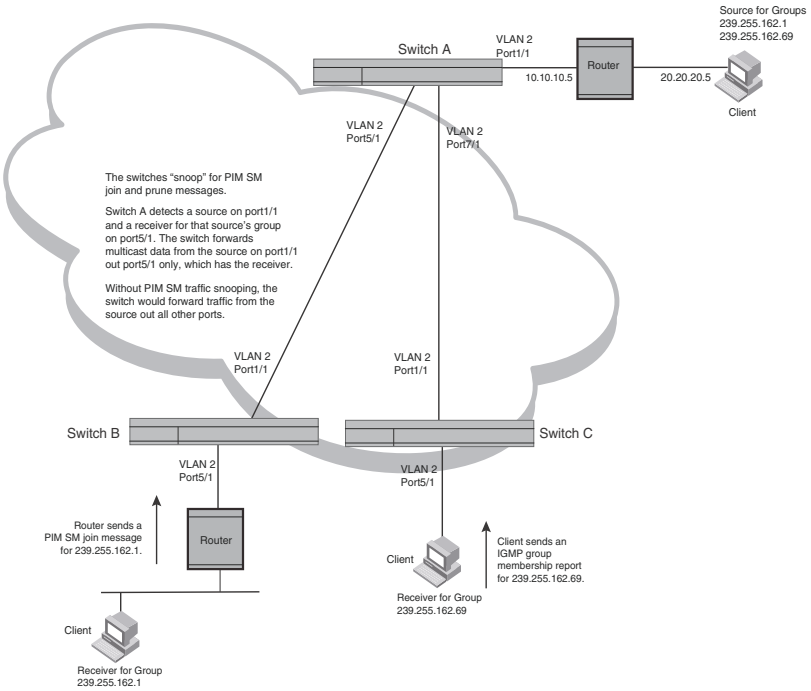
The IP multicast traffic reduction feature and the PIM SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM SM groups learned through join messages as well as MAC addresses learned through IGMP group membership reports. In this case, even 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 136.

Figure 137 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 *BigIron RX*.

FIGURE 137 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

To configure PIM SM traffic snooping:

- 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 *BigIron RX*, PIM SM traffic snooping will not be supported.

NOTE
Multicast protocols can only be applied to 1 physical interface. You must create multiple VLANs with both taggged and untagged ports and ve's under which you configure pim.

Enabling PIM SM traffic snooping

To enable PIM SM traffic snooping, enter the following commands at the global CONFIG level of the CLI.

```
BigIron RX(config)# ip multicast
BigIron RX(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.

```
BigIron RX(config)# no ip pimsm-snooping
```

If you also want to disable IP multicast traffic reduction, enter the following command.

```
BigIron RX(config)# no ip multicast
```

Multicast traffic reduction per VLAN

With release 02.6.00 of the *Multi-Service IronWare* software, you can configure specified VLANs instances for multicast traffic reduction by these methods as described in the following sections. Additionally, release 02.6.00 introduces PIM proxy which are only configurable per VLAN instance.

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.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# multicast pimsm-snooping
```

Syntax: [no] multicast pimsm-snooping

Configuring PIM proxy per VLAN instance

Using the PIM proxy function, multicast traffic can be reduced by configuring an *BigIron RX* 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 an *BigIron RX* switch to function as a PIM proxy on VLAN 2, use the following commands.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# multicast pim-proxy-enable
```

Syntax: [no] multicast pim-proxy-enable

Displaying IP multicast information

The following sections show how to display and clear IP multicast reduction information.

Displaying multicast information

To display IP multicast traffic reduction information on the *BigIron RX*, enter the following command at any level of the CLI.

```
BigIron RX(config)# show ip multicast
IP multicast is enabled - Passive
IP pimsm snooping is enabled

VLAN ID 23
Active 10.10.10.10 Report ports: 1/1 7/1
Report FID 0X0400
Number of Multicast Groups: 2

1     Group: 225.1.0.291
      IGMP report ports  :
      Mapped mac address : 0100.5e01.001d Fid:0x041b
      PIMv2*G join ports : 1/1

2     Group: 225.1.0.24
      IGMP report ports : 4/48
      Mapped mac address : 0100.5e01.0018 Fid:0x041a
      PIMv2*G join ports : 1/1
```

Syntax: show ip multicast

This display shows the following information.

This field...	Displays...
IP multicast traffic snooping state	Indicates whether IP multicast traffic snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
IP PIMSM snooping state	Indicates if PIM snooping is enabled. If disabled, this line does not appear.
VLAN ID	The port-based VLAN to which the information listed below the VLAN ID applies. Each port-based VLAN is a separate Layer 2 broadcast domain.
Active	The IP address of the device that actively sends IGMP queries.
Router Ports	The ports that are connected to routers that support IP multicast.
Report FID	The fid and camindex values are used by *Brocade* Technical Support for troubleshooting.
Number of Multicast Group	The total number of groups for which the VLAN's ports have received IGMP group membership reports, join messages, or prune messages.
Group	An IP multicast group.
IGMP Report Port	The ports in this VLAN on which the *BigIron RX* has received IGMP group membership reports for IP multicast groups. This line is blank if PIM snooping is enabled.
PIMv2 *G join ports	Ports participating in PIM snooping. This line is not displayed if PIM snooping is disabled.

Displaying IP multicast statistics

To display IP multicast statistics on a device, enter the following commands at any level of the CLI.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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

BigIron RX# clear ip multicast all

BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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

BigIron RX# clear ip multicast group 239.255.162.5

BigIron RX# 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.

Chapter

42

IPv6 Addressing

In this chapter

This chapter includes overview information about the following topics:

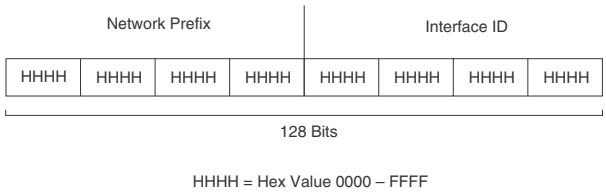
- IPv6 addressing.
- The IPv6 stateless autoconfiguration feature, which enables a host on a local link to automatically configure its interfaces with new and globally unique IPv6 addresses associated with its location.

IPv6 addressing

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 138 shows the IPv6 address format.

FIGURE 138 IPv6 address format



As shown in Figure 138, 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 138, 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*>/<*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 switch interface. Table 165 presents the three major types of IPv6 addresses that you can assign to a switch 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 165 describes global, site-local, and link-local addresses and the topologies in which they are used.
- Multicast addresses support a scope field, which Table 165 describes.

TABLE 165 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 switch 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 switch only. An anycast address must not be used as the source address of an IPv6 packet.

A switch 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.

IPv6 stateless autoconfiguration

Brocade routers use the IPv6 stateless autoconfiguration 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 switch on a local link periodically sends switch 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.

NOTE
For the stateless auto configuration feature to work properly, the advertised prefix length in switch advertisement messages must always be 64 bits.

The IPv6 stateless autoconfiguration 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 switch 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 switch advertisements. At this point, only addresses that contain the new prefix are used on the link. .

Chapter

43

Configuring Basic IPv6 Connectivity

In this chapter

This chapter explains how to get a *Brocade Layer 3 Switch* that supports IPv6 up and running. To configure basic IPv6 connectivity, you must do the following:

- Enable IPv6 routing globally on the *Brocade Layer 3 Switch*.
- 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 ECMP Load Sharing for IPv6
- Configure IPv6 ICMP.
- Configure the IPv6 neighbor discovery feature.
- Change the IPv6 MTU.
- Configure an unnumbered interface.
- 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 disabled. To enable the forwarding of IPv6 traffic globally on the router, enter the following command.

```
BigIron RX(config)# ipv6 unicast-routing
```

Syntax: [no] ipv6 unicast-routing

To disable the forwarding of IPv6 traffic globally on the *Brocade* device, enter the **no** form of this command.

Configuring IPv6 on each router interface

To forward IPv6 traffic on a router interface, the interface must have an IPv6 address, or IPv6 must be explicitly enabled. By default, an IPv6 address is not configured on a router 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 on the interface, 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 1084.

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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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: 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 a router interface without configuring a global or site-local address for the interface, enter commands such as the following.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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, *Brocade* recommends 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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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: 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 belonging to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface configured with the 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 *Brocade* 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.

```
BigIron RX(config)# int e 2/1
BigIron RX(config-if-e100-2/1)# ipv6 address 2002::6/64 anycast
```

Syntax: ipv6 address <*ipv6-prefix*>/<*prefix-length*> [anycast]

IPv6 anycast addresses are described in detail in RFC 1884. Refer to 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 have the management port configured to automatically obtain an IPv6 address. This process is the same for any other port 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 1069

IPv6 host support

You can configure the device to be an IPv6 host. An ***IPv6 host*** has IPv6 addresses on its interfaces, but does not have IPv6 routing enabled on it. This feature support is available on devices running the Layer 2, base Layer 3, or full Layer 3 software image.

This section lists the 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. Manual IPv6 address configuration is not supported. Also, automatic configuration of an IPv6 global address is supported only if there is an IPv6 router present on the network.

- 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
- TACAS, RADIUS, NTP over IPv6

IPv6 CLI command support

Table 166 lists the IPv6 CLI commands supported.

TABLE 166 IPv6 CLI command support

IPv6 command	Description	Switch code	Router code
clear ipv6 cache	Deletes all entries in the dynamic host cache.		X
clear ipv6 neighbor	Deletes all dynamic entries in the IPv6 neighbor table.	X	X
clear ipv6 route	Deletes all dynamic entries in the IPv6 route table.		X
clear ipv6 traffic	Resets all IPv6 packet counters.	X	X
copy tftp	Downloads a copy of a *Brocade* software image from a TFTP server into the system flash using IPv6.	X	X
debug ipv6	Displays IPv6 debug information.	X	X
ipv6 access-class	Configures access control for IPv6 management traffic.	X	X
ipv6 access-list	Configures an IPv6 access list for IPv6 access control.	X	X
ipv6 dns domain-name	Configures an IPv6 domain name.	X	X
ipv6 dns server-address	Configures an IPv6 DNS server address.	X	X
ipv6 enable	Enables IPv6 on an interface.		X
ipv6 neighbor	Maps a static IPv6 address to a MAC address in the IPv6 neighbor table.		X
ipv6 route	Configures an IPv6 static route.		X
log host ipv6	Configures the IPv6 Syslog server. Refer to "Configuring an IPv6 Syslog server" on page 1074.	X	X
no ipv6 enable	Disables IPv6 on a global basis on a Layer 2 switch.	X	
ping ipv6	Performs an ICMP for IPv6 echo test.	X	X
show ipv6	Displays some global IPv6 parameters, such IPv6 DNS server address.	X	X
show ipv6 access-list	Displays configured IPv6 access lists.	X	X
show ipv6 cache	Displays the IPv6 host cache.		X
show ipv6 interface	Displays IPv6 information for an interface.		X

TABLE 166 IPv6 CLI command support (Continued)

IPv6 command	Description	Switch code	Router code
show ipv6 neighbor	Displays the IPv6 neighbor table.	X	X
show ipv6 route	Displays IPv6 routes.		X
show ipv6 router	Displays IPv6 local routers.		X
show ipv6 tcp	Displays information about IPv6 TCP sessions.	X	X
show ipv6 traffic	Displays IPv6 packet counters.	X	X
snmp-client ipv6	Restricts SNMP access to a certain IPv6 node. Refer to "Restricting SNMP access to an IPv6 node" on page 1073.	X	X
snmp-server host ipv6	Specifies the recipient of SNMP notifications. Refer to "Specifying an IPv6 SNMP trap receiver" on page 1073.	X	X
telnet	Enables a Telnet connection from the *Brocade* device to a remote IPv6 host using the console.	X	X
traceroute ipv6	Traces a path from the *Brocade* device to an IPv6 host.	X	X
web access-group ipv6	Restricts Web management access to certain IPv6 hosts as determined by IPv6 ACLs. Refer to "Restricting web management access to an IPv6 host by specifying an IPv6 ACL" on page 1074.	X	X
web client ipv6	Restricts Web management access to certain IPv6 hosts. Refer to "Restricting web management access to an IPv6 host" on page 1074.	X	X

Restricting SNMP access to an IPv6 node

You can restrict SNMP access (which includes IronView Network Manager) to the device to the IPv6 host whose IP address you specify. To do so, enter a command such as the following:

```
BigIron RX(config)# snmp-client ipv6 2001:efff:89::23
```

Syntax: snmp-client ipv6 <*ipv6-address*>

The <*ipv6-address*> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Specifying an IPv6 SNMP trap receiver

You can specify an IPv6 host as a trap receiver to ensure that all SNMP traps sent by the device will go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. To do so, enter a command such as the following.

```
BigIron RX(config)# snmp-server host ipv6 2001:efff:89::13
```

Syntax: snmp-server host ipv6 <*ipv6-address*>

The <*ipv6-address*> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Restricting web management access to an IPv6 host by specifying an IPv6 ACL

You can specify an IPv6 ACL that restricts Web management access to management functions on the device. For example.

```
BigIron RX(config)# access-list 12 deny host 2000:2383:e0bb::2/128 log
BigIron RX(config)# access-list 12 deny 30ff:3782::ff89/128 log
BigIron RX(config)# access-list 12 deny 3000:4828::fe19/128 log
BigIron RX(config)# access-list 12 permit any
BigIron RX(config)# web access-group ipv6 12
```

Syntax: web access-group ipv6 <*ipv6 ACL name*>

where <*ipv6 ACL name*> is a valid IPv6 ACL.

Restricting web management access to an IPv6 host

You can restrict Web management access to the device to the IPv6 host whose IP address you specify. No other device except the one with the specified IPv6 address can access the *Brocade* device's Web management interface. For example.

```
BigIron RX(config)# web client ipv6 3000:2383:e0bb::2/128
```

Syntax: web client ipv6 <*ipv6-address*>

The <*ipv6-address*> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Configuring an IPv6 Syslog server

To specify an IPv6 Syslog server, enter a command such as the following.

```
BigIron RX(config)# log host ipv6 2000:2383:e0bb::4/128
```

Syntax: log host ipv6 <*ipv6-address*> [<*udp-port-num*>]

The <*ipv6-address*> you specify 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

Some of the **show** commands display IPv6 addresses for IPv6 SNMP servers. The following shows an example output for the **show snmp server** command.

```
BigIron RX# 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 node on a link to discover the routers on the same link. By default, router advertisement and solicitation messages are permitted on the device. To disable these messages, configure an IPv6 access list that denies them. The following shows an example configuration.

Example

```
BigIron RX(config)# ipv6 access-list rtradvert
BigIron RX(config)# deny icmp any any router-advertisement
BigIron RX(config)# deny icmp any any router-solicitation
BigIron RX(config)# permit ipv6 any any
```

Disabling IPv6 on a Layer 2 switch

IPv6 is enabled by default in the Layer 2 switch code. If desired, you can disable IPv6 on a global basis on a device running the switch code. To do so, enter the following command at the Global CONFIG level of the CLI.

```
BigIron RX(config)# no ipv6 enable
```

Syntax: no ipv6 enable

To re-enable IPv6 after it has been disabled, enter **ipv6 enable**.

NOTE
IPv6 is disabled by default in the router code and must be configured on each interface that will support IPv6.

Configuring an IPv6 host address for a *BigIron RX* running a switch image

NOTE
This feature is only available on the *BigIron RX* when it is configured as a switch. For this feature to work it must have the CHD code enabled on the *BigIron RX*.

In the router configuration, each port can be configured separately with an IPv6 address. This is accomplished using the interface configuration process that is described in "Configuring IPv6 on each router interface" on page 1068.

When a *BigIron RX* is running a switch-only image of the code, individual ports cannot be configured with an IP address (IPv4 or IPv6). In this situation the *BigIron RX* has one IP address for the management port, and one IP address for the system. This has previously been supported for IPv4 but not IPv6.

There is support for configuring an IPv6 address on the management port as described in "Configuring the management port for an IPv6 automatic address configuration" on page 1071 and for configuring a system-wide IPv6 address on the *BigIron RX* in switch mode. Configuration of the system-wide IPv6 address is exactly like configuration of an IPv6 address in router mode except that all of the IPv6 configuration is at the Global Config level instead of at the Interface Config level.

The process for defining the system-wide interface for IPv6 is described in the following sections:

- "Configuring a global or site-local IPv6 address with a manually configured interface ID as the switch's system-wide address" on page 1077
- "Configuring a global or site-local IPv6 address with an automatically computed EUI-64 interface ID as the switch's system-wide address" on page 1077
- Refer to the *"Configuring a Link-Local IPv6 Address as the Switch's System-Wide Address "* in the *BigIron RX Installation Guide*.

Configuring a global or site-local IPv6 address with a manually configured interface ID as the switch's system-wide address

To configure a global or site-local IPv6 Address with a manually configured interface ID, as a switch's system-wide address, enter a command such as the following at the Global Config level.

```
BigIron RX(config)#ipv6 address 2001:200:12D:1300:240:D0FF:FE48:4000:1/64
```

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 in 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 as the switch's system-wide address

To configure a global or site-local IPv6 address with an automatically computed EUI-64 interface ID in the low order 64-bits s the system-wide address, enter commands such as the following.

```
BigIron RX(config)# ipv6 address 2001:200:12D:1300::/64 eui-64
```

These commands configure the global prefix 2001:200:12d:1300::/64 and an interface ID as the system-wide address, and enable IPv6.

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.

Configuring a link-local IPv6 address as the switch's system-wide address

To enable IPv6 and automatically configure a global interface enter commands such as the following.

```
BigIron RX(config)# ipv6 enable
```

This command enables IPv6 on the switch and specifies that the interface is assigned an automatically computed link-local address.

Syntax: [no] ipv6 enable

To override a link-local address that is automatically computed for the global interface with a manually configured address, enter a command such as the following.

```
BigIron RX(config)# ipv6 address FE80::240:D0FF:FE48:4672 link-local
```

This command explicitly configures the link-local address FE80::240:D0FF:FE48:4672 for the global interface.

Syntax: 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 IPv4 and IPv6 protocol stacks

One situation in which you must configure a router to run both IPv4 and IPv6 protocol stacks is if it is deployed as an endpoint for an IPv6 over IPv4 tunnel.

Each router interface that you want to send and receive both IPv4 and IPv6 traffic must be configured with an IPv4 address and an IPv6 address. (An alternative to configuring a router interface with an IPv6 address is to explicitly enable IPv6 using the **ipv6 enable** command. For more information about using this command, refer to "Configuring a link-local IPv6 address" on page 1069.)

To configure a router interface to support both the IPv4 and IPv6 protocol stacks, use commands such as the following.

```
BigIron RX(config)# ipv6 unicast-routing
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e100-3/1)# ip address 192.168.1.1 255.255.255.0
BigIron RX(config-if-e100-3/1)# ipv6 address 2001:200:12d:1300::/64 eui-64
```

These commands globally enable IPv6 routing on the router 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: 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: 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 1069.

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. 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 1068.

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 *Brocade* device and thereby recognize all hosts within that domain. After you define a domain name, the *Brocade* 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 *Brocade* device, 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.

```
BigIron RX# ping nyc01
BigIron RX# 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 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 *Brocade* device and then define four possible default DNS gateway addresses. To do so using IPv4 addressing, you would enter the following commands.

```
BigIron RX(config)# ip dns domain-name newyork.com
BigIron RX(config)# ip dns server-address 209.157.22.199 205.96.7.15 208.95.7.25
201.98.7.15
```

Syntax: 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.

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. *Brocade* 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. They store a complete IPv6 address in each record. AAAA records have a type value of 28.

To establish an IPv6 DNS entry for the device, enter the following command.

```
BigIron RX(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.

```
BigIron RX(config)# ipv6 dns server-address 200::1
```

Syntax: [no] ipv6 dns server-address <*ipv6-addr*> [<*ipv6-addr*>] [<*ipv6-addr*>] [<*ipv6-addr*>]

As an example, in a configuration where ftp6.companynet.com is a server with an IPv6 protocol stack, when a user pings ftp6.companynet.com, the *Brocade* 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.

ECMP load sharing for IPv6

The IPv6 route table selects the best route to a given destination from among the routes in the tables maintained by the configured routing protocols (BGP4, OSPF, static, and so on). The IPv6 route table can contain more than one path to a given destination. When this occurs, the *Brocade* device selects the path with the lowest cost for insertion into the routing table. If more than one path with the lowest cost exists, all of these paths are inserted into the routing table, subject to the configured maximum number of load sharing paths (by default 4). The device uses ***Equal-Cost Multi-Path (ECMP) load sharing*** to select a path to a destination.

When the device receives traffic for a destination, and the IPv6 route table contains multiple, equal-cost paths to that destination, the device checks the ***IPv6 forwarding cache*** for a forwarding entry for the destination. The IPv6 forwarding cache provides a fast path for forwarding IPv6 traffic. The IPv6 forwarding cache contains entries that associate a destination host or network with a path (next-hop router).

If the IPv6 forwarding cache contains a forwarding entry for the destination, the *Brocade* device uses the entry to forward the traffic. If the IPv6 forwarding cache does not contain a forwarding entry for the destination, the software selects a path from among the available equal-cost paths to the destination, then creates an entry in the in the cache based on the calculation. Subsequent traffic for the same destination uses the forwarding entry. Entries remain in the IPv6 forwarding cache for one minute, then are aged out.

If the path selected by the device becomes unavailable, its entry in the IPv6 forwarding cache is removed, a new path is selected from the remaining equal-cost paths to the destination, and an entry is created in the IPv6 forwarding cache using the new path.

Brocade devices support the following ECMP load-sharing methods for IPv6 traffic:

- **Network-based** – The *Brocade* device distributes traffic across equal-cost paths based on destination network address. The software selects a path based on a calculation involving the maximum number of load-sharing paths allowed and the actual number of paths to the destination network. This is the default ECMP load-sharing method for IPv6.
- **Host-based** – The *Brocade* device uses a simple round-robin mechanism to distribute traffic across the equal-cost paths based on destination host IP address. The device uses this ECMP load-sharing method for IPv6 if you explicitly configure it to do so.

You can manually disable or enable ECMP load sharing for IPv6, specify the number of equal-cost paths the device can distribute traffic across, and configure the device to use the host-based ECMP load-sharing method instead of the network-based method. In addition, you can display information about the status of ECMP load-sharing on the device, as well as the entries in the IPv6 forwarding cache.

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.

```
BigIron RX(config)# no ipv6 load-sharing
```

If you want to re-enable the feature after disabling it, you must specify the number of load-sharing paths. The maximum number of paths the device supports is a value from 2 – 8. By entering a command such as the following, iPv6 load-sharing will be re-enabled.

```
BigIron RX(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. The default is 4..

Changing the maximum number of load sharing paths for IPv6

By default, IPv6 ECMP load sharing allows traffic to be balanced across up to four equal paths. You can change the maximum number of paths the device supports to a value from 2 – 8.

To change the number of ECMP load sharing paths for IPv6, enter a command such as the following.

```
BigIron RX(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.

Changing the ECMP load-sharing method for IPv6

Brocade devices can perform ECMP load-sharing for IPv6 traffic based on destination host address or destination network. The default is network-based IP load sharing. If you want to enable the device to perform host-based IP load sharing instead, enter the following command.

```
BigIron RX(config)# ipv6 load-sharing by-host
```

Syntax: [no] ipv6 load-sharing by-host

This command enables host-based ECMP load sharing on the device. The command also disables network-based ECMP load-sharing at the same time.

DHCP relay agent for IPv6

A client locates a DHCP server using a reserve, link-scoped multicast address. For this reason, it is a requirement for direct communication between the client and the server that they be attached by the same link. However, in some situations in which ease of management, economy, and scalability is a concern, it is useful to allow a DHCP client to send a message to a DHCP server by using a DHCP relay agent. A DHCP relay agent, which may reside on the clients link, is used to relay messages between the client and the server. A DHCP relay agent is transparent to the client.

When the relay agent receives a message to be relayed from a client to another relay agent, it creates a new Relay-forward message, puts the original DHCP message to relay forward option, and includes its only address and the address it received is in the same option.

Configuring DHCP for IPv6 relay agent

You can enable the DHCP for IPv6 relay agent function and specify the relay destination addresses on an interface by entering the command at the interface level.

```
BigIron(config)# interface ethernet 2/3
BigIron(config-if-e10000-2/3)#ipv6 dhcp-relay-dest 2001::2
```

Syntax: ipv6 dhcp-relay-dest < *ipv6-address* >

Select the **ipv6-address** to specify a destination address to which the client messages are forwarded and enables DHCP for IPv6 relay services on the interface.

Enabling support for network-based ECMP load sharing for IPv6

Network-based ECMP load sharing is supported. If this configuration is selected, 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 select network-based ECMP load sharing, you can choose either of the following two CAM modes:

- **Dynamic Mode** – In the dynamic mode, routes are entered into the CAM dynamically using a flow-based scheme. In this mode routes are only added to the CAM as they are required. Once routes are added to the CAM, they are subject to being 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 they are subject to being aged-out when they are not in use.

Displaying ECMP load-sharing information for IPv6

To display the status of ECMP load sharing for IPv6, enter the following command.

```
BigIron RX# show ipv6
Global Settings
  unicast-routing enabled, hop-limit 64
  No Inbound Access List Set
  No Outbound Access List Set
  Prefix-based IPv6 Load-sharing is Enabled, Number of load share paths: 4
```

Syntax: show ipv6

You can display the entries in the IPv6 forwarding cache; for example:

```
BigIron RX# 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

As with the Internet Control Message Protocol (ICMP) for IPv4, ICMP for IPv6 provides error and informational messages. *Brocade*'s implementation of the stateless autoconfiguration, neighbor discovery, and path MTU discovery features use ICMP messages.

This section explains how to configure the following IPv6 ICMP features:

- ICMP rate limiting.
- ICMP redirects.

Configuring ICMP rate limiting

You can limit the rate at which IPv6 ICMP error messages are sent out on a network. IPv6 ICMP implements a token bucket algorithm.

To illustrate how this algorithm works, imagine a virtual bucket that contains a number of tokens. Each token represents the ability to send one ICMP error message. Tokens are placed in the bucket at a specified interval until the maximum number of tokens allowed in the bucket is reached. For each error message that ICMP sends, a token is removed from the bucket. If ICMP generates a series of error messages, messages can be sent until the bucket is empty. If the bucket is empty of tokens, 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.

```
BigIron RX(config)# ipv6 icmp error-interval 1000 100
```

Syntax: ipv6 icmp error-interval <*interval*> [<*number-of-tokens*>]

The interval in milliseconds at which tokens are placed in the bucket can range from 0 – 2147483647. The maximum number of tokens stored in the bucket can range from 1 – 200.

NOTE
If you retain the default interval value or explicitly set the value to 100 milliseconds, output from the **show run** command does not include the setting of the **ipv6 icmp error-interval** command because the setting is the default.

Also, if you configure the interval value to a number that does not evenly divide into 100000 (100 milliseconds), the system rounds up the value to a next higher value that does divide evenly into 100000. For example, if you specify an interval value of 150, the system rounds up the value to 200.

ICMP rate limiting is enabled by default. To disable ICMP rate limiting, set the interval to zero.

Disabling or reenabling ICMP redirect messages

You can disable or re-enable the sending of ICMP redirect messages by a router. By default, a router can send an ICMP redirect message to a neighboring host to inform it of a better first-hop router 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 1084.)

For example, to disable the sending of ICMP redirect messages on Ethernet interface 3/1, enter the following commands.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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.

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 routers.

An IPv6 host is required to listen for and recognize the following addresses that identify itself:

- 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 router is advertised as a default router (for use by all nodes on a given link).
  - Prefixes advertised in router advertisement messages.
  - Flags for host stateful autoconfiguration.
- Amount of time during which an IPv6 node considers a remote node reachable (for use by all nodes on a given link).

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 routers on the same link.

Each configured router 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 router 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 router Ethernet interfaces. You can configure several router advertisement message parameters. For information about disabling the sending of router advertisement messages and the router advertisement parameters that you can configure, refer to "Enabling and disabling IPv6 router advertisements" on page 1089 and "Setting IPv6 router advertisement parameters" on page 1087.

Neighbor redirect messages

After forwarding a packet, by default, a router can send a neighbor redirect message to a host to inform it of a better first-hop router. The host receiving the neighbor redirect message will then readdress the packet to the better router.

A router 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 *Brocade* 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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e100-3/1)# ipv6 nd dad attempt 2
BigIron RX(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. *Brocade* does not recommend very short intervals in normal IPv6 operation. When a non-default value is configured, the configured time is both advertised and used by the router 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 router is advertised as a default router on this interface. If you set the value of this parameter to 0, the router is not advertised as a default router on an interface. If you set this parameter to a value that is not 0, the router is advertised as a default router on this interface. By default, the router lifetime value included in router advertisement messages sent from an interface is 1800 seconds.

When adjusting these parameter settings, *Brocade* recommends that the interval between router advertisement transmission be less than or equal to the router lifetime value if the router is advertised as a default router. For example, to adjust the interval of router advertisements to 300 seconds and the router lifetime value to 1900 seconds on Ethernet interface 3/1, enter the following commands.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e100-3/1)# ipv6 nd ra-interval 300
BigIron RX(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 router 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 router 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 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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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 nonaddress 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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e100-3/1)# ipv6 nd managed-config-flag
BigIron RX(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.

Enabling and disabling IPv6 router advertisements

If IPv6 unicast routing is enabled on an Ethernet interface, by default, this interface sends IPv6 router advertisement messages. However, by default, non-LAN interface types, for example, tunnel interfaces, do not send router advertisement messages.

To disable the sending of router advertisement messages on an Ethernet interface, enter commands such as the following.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e100-3/1)# ipv6 nd suppress-ra
```

To enable the sending of router advertisement messages on a tunnel interface, enter commands such as the following.

```
BigIron RX(config)# interface tunnel 1
BigIron RX(config-tnif-1)# no ipv6 nd suppress-ra
```

Syntax: [no] ipv6 nd suppress-ra

Configuring reachable time for remote IPv6 nodes

You can configure the duration (in seconds) that a router considers a remote IPv6 node reachable. By default, a router interface uses the value of 30 seconds.

The router advertisement messages sent by a router 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
For the interval at which a router interface sends router advertisement messages, *Brocade* uses seconds as the unit of measure instead of milliseconds.

Brocade does not recommend configuring 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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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.

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 fragments the packet and transmits its contents 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.

For example, to configure the MTU on Ethernet interface 3/1 as 1280 bytes, enter the following commands.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e100-3/1)# ipv6 mtu 1280
```

Syntax: [no] ipv6 mtu <*bytes*>

You can specify between 1280 – 1500 bytes. If a nondefault value is configured for an interface, router advertisements include an MTU option.

You can configure IPv6 MTU for to be greater than 1500 bytes, although the default remains at 1500 bytes. The value of the MTU you can define depends on the following:

- For a physical port, the maximum value of the MTU is the equal to the maximum frame size of the port minus 18 (Layer 2 MAC header + CRC).
- For a virtual routing interface, the maximum value of the MTU is the maximum frame size configured for the VLAN to which it is associated, minus 18 (Layer 2 MAC header + CRC). If a maximum frame size for a VLAN is not configured, then configure the MTU based on the smallest maximum frame size of all the ports of the VLAN that corresponds to the virtual routing interface, minus 18 (Layer 2 MAC header + CRC).

To define IPv6 MTU globally, enter the following command.

```
BigIron RX(config)#ipv6 mtu 1300
```

To define IPv6 MTU on an interface, enter the following command:

```
BigIron RX(config-if-e1000-2/1)#ipv6 mtu
```

Syntax: ipv6 mtu <*value*>

NOTE
If a the size of a jumbo packet received on a port is equal to the maximum frame size – 18 (Layer 2 MAC header + CRC) and if this value is greater than the outgoing port's IPv4/IPv6 MTU, then it will be forwarded in the CPU.

Configuring static neighbor entries

In some special cases, a neighbor cannot be reached using the neighbor discovery feature. 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.

```
BigIron RX(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, specify the port number of the Ethernet interface. 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, you can enter the following command.

```
BigIron RX(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 *Brocade* device apply to both incoming IPv6 and IPv4 traffic. ACLs can be used to perform QoS for IPv6 traffic:

- dscp

- fragments
- priority-force
- priority-mapping
- source routing

To enable QoS for IPv6 traffic, enter the following commands.

```
BigIron RX(config)# port-priority
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# 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 the *BigIron RX* routers, when QoS is enabled with the **port-priority** command, the device inserts a value in the internal *Brocade* 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

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.

```
BigIron RX# clear ipv6 cache 2000:e0ff::1
```

Syntax: clear ipv6 cache [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | ethernet <*port*> | tunnel <*number*> | 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 | tunnel | ve** parameter specifies the interfaces for which you can remove cache entries. 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, respectively.

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.

```
BigIron RX# 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, also specify the port number associated with the interface. If you specify a VE, 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.

```
BigIron RX# 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.

```
BigIron RX(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.

```
BigIron RX# 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 that has indices to the next hop gateway and the router interface on which the route was learned.

To display IPv6 cache information, enter the following command at any CLI level:

```
BigIron RX# 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*> | ve <*number*> | tunnel <*number*>]

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 | 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.

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. If you specify a tunnel interface, also specify the tunnel number.

This display shows the following information.

TABLE 167 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 router. • Local – The next hop is originated on this router. • <*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.

```
BigIron RX# 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, 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 168 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.

To display detailed information for a specific interface, enter a command such as the following at any CLI level.

```
BigIron RX# 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
```

This display shows the following information.

TABLE 169 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 address(es), if one or more are configured for the interface.
Joined group address(es)	The multicast address(es) that a router interface listens for and recognizes.

TABLE 169 Detailed IPv6 interface information fields (Continued)

This field...	Displays...
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.

Displaying IPv6 neighbor information

You can display the IPv6 neighbor table, which contains an entry for each IPv6 neighbor with which the router exchanges IPv6 packets.

To display the IPv6 neighbor table, enter the following command at any CLI level.

```
BigIron RX(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, also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

This display shows the following information.

TABLE 170 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.

TABLE 170 IPv6 neighbor information fields (Continued)

This field...	Displays...
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 router or host. **0** – Indicates that the neighbor is a host. **1** – Indicates that the neighbor is a router.

Displaying the IPv6 route table

To display the IPv6 route table, enter the following command at any CLI level.

```
BigIron RX# show ipv6 route
IPv6 Routing Table - 7 entries:
Type Codes:  C - Connected, S - Static, R - RIP, O - OSPF, B - BGP, I - ISIS
Type IPv6 Prefix                Next Hop Router           Interface  Dis/Metric
C  2000:4::/64                  ::                        ethe 3/2    0/0
S  2002::/16                    ::                        tunnel 6    1/1
S  2002:1234::/32               ::                        tunnel 6    1/1
C  2002:c0a8:46a::/64           ::                        ethe 3/2    0/0
C  2999::1/128                  ::                        loopback 2  0/0
O  2999::2/128                  fe80::2e0:52ff:fe91:bb37  ethe 3/2    110/1
C  5000:2::/64                  ::                        tunnel 2    0/0
```

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 171 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: • C – The destination is directly connected to the router. • S – The route is a static route. • R – The route is learned from RIPng. • O – The route is learned from OSPFv3. • B – The route is learned from BGP4+. • I – The route is learned from IPv6 IS-IS.
IPv6 Prefix	The destination network of the route.
Next-Hop Router	The next-hop router.
Interface	The interface through which this router sends packets to reach the route's destination.
Dis/Metric	The route's administrative distance and metric value.

To display a summary of the IPv6 route table, enter the following command at any CLI level.

```
BigIron RX# 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
```

The following table lists the information displayed by the **show ipv6 route summary** command.

TABLE 172 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 routers

The *Brocade* device can function as an IPv6 host, instead of an IPv6 router, if you configure IPv6 addresses on its interfaces but do not enable IPv6 routing using the **ipv6 unicast-routing** command.

From the IPv6 host, you can display information about IPv6 routers to which the host is connected. The host learns about the routers through their router advertisement messages. To display information about the IPv6 routers connected to an IPv6 host, enter the following command at any CLI level.

```
BigIron RX# 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 *Brocade* device to function as an IPv6 router (you configure IPv6 addresses on its interfaces and enable IPv6 routing using the **ipv6 unicast-routing** command) and you 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 *Brocade* devices configured to function as IPv6 hosts only.

This display shows the following information.

TABLE 173 IPv6 local router information fields

This field...	Displays...
Router <*ipv6 address*> on <*interface*> <*port*>	The IPv6 address for a particular router interface.
Last update	The amount of elapsed time (in minutes) between the current and previous updates received from a router.
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 router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router leaves this field unspecified.
Lifetime	The amount of time (in seconds) that the router is useful as the default router.
Reachable time	The amount of time (in milliseconds) that a router assumes a neighbor is reachable after receiving a reachability confirmation. The reachable time value applies to the router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router 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 router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router leaves this field unspecified.

Displaying IPv6 TCP information

You can display the following IPv6 TCP information:

- General information about each TCP connection on the router, 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 router, enter the following command at any CLI level.

```
BigIron RX# 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 174 General IPv6 TCP connection fields

This field...	Displays...
Local IP address:port	The IPv4 or IPv6 address and port number of the local router interface over which the TCP connection occurs.
Remote IP address:port	The IPv4 or IPv6 address and port number of the remote router interface over which the TCP connection occurs.
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 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.
FREE TCB = <*percentage*>	The percentage of free TCP control block (TCB) space.

TABLE 174 General IPv6 TCP connection fields (Continued)

This field...	Displays...
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.

```
BigIron RX# 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 175 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 174 on page 1101. • The port numbers of the local interface.
Send: initial sequence number = <*number*>	The initial sequence number sent by the local router.
Send: first unacknowledged sequence number = <*number*>	The first unacknowledged sequence number sent by the local router.
Send: current send pointer = <*number*>	The current send pointer.
Send: next sequence number to send = <*number*>	The next sequence number sent by the local router.
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 router.
Send: total used buffers <*number*>	The total number of buffers used by the local router in setting up the TCP connection.
Receive: initial incoming sequence number = <*number*>	The initial incoming sequence number received by the local router.
Receive: expected incoming sequence number = <*number*>	The incoming sequence number expected by the local router.
Receive: received window = <*number*>	The size of the local router's receive window.
Receive: bytes in receive queue = <*number*>	The number of bytes in the local router's receive queue.
Receive: congestion window = <*number*>	The size of the local router's receive congestion window.

Displaying IPv6 traffic statistics

To display IPv6 traffic statistics, enter the following command at any CLI level.

```
BigIron RX# 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 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 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 176 IPv6 traffic statistics fields

This field...	Displays...
IPv6 statistics	
received	The total number of IPv6 packets received by the router.
sent	The total number of IPv6 packets originated and sent by the router.
forwarded	The total number of IPv6 packets received by the router and forwarded to other routers.
delivered	The total number of IPv6 packets delivered to the upper layer protocol.
rawout	This information is used by *Brocade* Technical Support.
bad vers	The number of IPv6 packets dropped by the router because the version number is not 6.

TABLE 176 IPv6 traffic statistics fields (Continued)

This field...	Displays...
bad scope	The number of IPv6 packets dropped by the router because of a bad address scope.
bad options	The number of IPv6 packets dropped by the router because of bad options.
too many hdr	The number of IPv6 packets dropped by the router because the packets had too many headers.
no route	The number of IPv6 packets dropped by the router because there was no route.
can't forward	The number of IPv6 packets the router could not forward to another router.
redirect sent	This information is used by *Brocade* Technical Support.
frag recv	The number of fragments received by the router.
frag dropped	The number of fragments dropped by the router.
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 router reassembled.
fragmented	The number of IPv6 packets fragmented by the router to accommodate the MTU of this router or of another device.
ofragments	The number of output fragments generated by the router.
can't frag	The number of IPv6 packets the router 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	
dest unreach	The number of Destination Unreachable messages sent or received by the router.
pkt too big	The number of Packet Too Big messages sent or received by the router.
time exceeded	The number of Time Exceeded messages sent or received by the router.
param prob	The number of Parameter Problem messages sent or received by the router.
echo req	The number of Echo Request messages sent or received by the router.
echo reply	The number of Echo Reply messages sent or received by the router.
mem query	The number of Group Membership Query messages sent or received by the router.
mem report	The number of Membership Report messages sent or received by the router.
mem red	The number of Membership Reduction messages sent or received by the router.

TABLE 176 IPv6 traffic statistics fields (Continued)

This field...	Displays...
router soli	The number of Router Solicitation messages sent or received by the router.
router adv	The number of Router Advertisement messages sent or received by the router.
nei soli	The number of Neighbor Solicitation messages sent or received by the router.
nei adv	The number of Router Advertisement messages sent or received by the router.
redirect	The number of redirect messages sent or received by the router.
Applies to received only	
bad code	The number of Bad Code messages received by the router.
too short	The number of Too Short messages received by the router.
bad checksum	The number of Bad Checksum messages received by the router.
bad len	The number of Bad Length messages received by the router.
nd toomany opt	The number of Neighbor Discovery Too Many Options messages received by the router.
badhopcount	The number of Bad Hop Count messages received by the router.
Applies to sent only	
error	The number of Error messages sent by the router.
can't send error	The number of times the node encountered errors in ICMP error messages.
too freq	The number of times the node 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 router.
admin	The number of Admin errors sent by the router.
beyond scope	The number of Beyond Scope errors sent by the router.
address	The number of Address errors sent by the router.
no port	The number of No Port errors sent by the router.
pkt too big	The number of Packet Too Big errors sent by the router.
time exceed transit	The number of Time Exceed Transit errors sent by the router.
time exceed reassembly	The number of Time Exceed Reassembly errors sent by the router.
param problem header	The number of Parameter Problem Header errors sent by the router.
nextheader	The number of Next Header errors sent by the router.
option	The number of Option errors sent by the router.
redirect	The number of Redirect errors sent by the router.
unknown	The number of Unknown errors sent by the router.
UDP statistics	
received	The number of UDP packets received by the router.
sent	The number of UDP packets sent by the router.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.

TABLE 176 IPv6 traffic statistics fields (Continued)

This field...	Displays...
input errors	This information is used by *Brocade* Technical Support.
TCP statistics	
active opens	The number of TCP connections opened by the router by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by the router in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by *Brocade* Technical Support.
active resets	The number of TCP connections the router 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 router reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by *Brocade* Technical Support.
in segments	The number of TCP segments received by the router.
out segments	The number of TCP segments sent by the router.
retransmission	The number of segments that the router 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

If you want to display the contents of an IPv6 session cache, enter the following command.

```
BigIron RX# 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.

Example

To show all IPv6 flows, enter the following command.

```
BigIron RX# show ipv6 flows
```

To show all IPv6 flows with any IPv6 source and any IPv6 destination addresses, enter the following command.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# show ipv6 flows host 4050::30 host 5020::30
```

The following is an example of what is displayed when you enter the **show ipv6 flows** command.

```
BigIron RX# 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/UDP port or the ICMP type of the flow.
- **DestPort/IcmpCode** – Either the destination TCP/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

44

Configuring RIPng

In this chapter

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, Brocade implements some new commands that are specific to RIPng. 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 RIPng" on page 1109

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 Brocade device and on individual router interfaces

The following configuration tasks are optional:

- Change the default settings of RIPng timers
- Configure how the Brocade device learns and advertises routes
- Configure which routes are redistributed into RIPng from other sources
- Configure how the Brocade device distributes routes through RIPng
- Configure poison reverse parameters

Enabling RIPng

Before configuring the Brocade device to run RIPng, you must do the following:

- Enable the forwarding of IPv6 traffic on the Brocade 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 task, refer to Chapter 43, "Configuring Basic IPv6 Connectivity".

By default, RIPng is disabled. To enable RIPng, you must enable it globally on the Brocade device and also on individual router interfaces.

NOTE
You are required to configure a router ID when running only IPv6 routing protocols.

NOTE
Enabling RIPng globally on the Brocade device does not enable it on individual router interfaces.

To enable RIPng globally, enter the following command.

```
BigIron RX(config-rip-router)#ipv6 router rip
BigIron RX(config-ripng-router)#
```

After you enter this command, the Brocade 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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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 177 describes the RIPng timers and provides their defaults.

TABLE 177 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.

TABLE 177 RIPng timers (Continued)

Timer	Description	Default
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, Brocade strongly recommends setting the same timer values for all routers and access servers in the network.
- Setting the update timer to a shorter interval can cause the routers to spend excessive time updating the IPv6 route table.
- Brocade recommends setting the timeout timer value to at least three times the value of the update timer.
- Brocade recommends 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.

```
BigIron RX(config)# ipv6 router rip
BigIron RX(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 Brocade 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:

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e100-3/1)# ipv6 address 2001:469e:0:adff:8935:e838:78:
e0ff /64
BigIron RX(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 Brocade device then places the route in the route table. When the Brocade 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.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(config-if-e100-3/1)# ipv6 rip metric-offset 1
BigIron RX(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 Brocade 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 Brocade device can use RIPng to advertise the route to its RIPng neighbors.

When configuring the Brocade 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.

```
BigIron RX(config)# ipv6 router rip
BigIron RX(config-ripng-router)# redistribute ospf
```

Syntax: 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.

```
BigIron RX(config)# ipv6 prefix-list routesfor2001 permit 2001::/16
BigIron RX(config)# ipv6 router rip
BigIron RX(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.

```
BigIron RX(config)# ipv6 prefix-list 3ee0routes deny 3ee0:a99::/64 le 128
BigIron RX(config)# ipv6 prefix-list 3ee0routes permit ::/0 ge 0 le 128
BigIron RX(config)# ipv6 router rip
BigIron RX(config-ripng-router)# distribute-list prefix-list 3ee0routes in
tunnel 1
```

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**, **loopback**, **ve**, or **tunnel** keywords. 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.

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.

```
BigIron RX(config)# ipv6 router rip
BigIron RX(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 Brocade 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.

```
BigIron RX(config)# ipv6 router rip
BigIron RX(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.

```
BigIron RX# 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.

```
BigIron RX# 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 178 RIPng configuration fields

This field...	Displays...
IPv6 RIP status/port	The status of RIPng on the Brocade 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/expiration	The settings of the RIPng update and timeout timers.
Holddown/garbage collection	The settings of the RIPng hold-down and garbage-collection timers.

TABLE 178 RIPng configuration fields (Continued)

This field...	Displays...
Split horizon/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/trigger updates	The number of periodic updates and triggered updates sent by the RIPng router.
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.

```
BigIron RX# 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 179 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*>	The IPv6 prefix and prefix length.
<*ipv6-address*>	The IPv6 address.

TABLE 179 RIPng routing table fields (Continued)

This field...	Displays...
Next-hop router	The next-hop router for this Brocade device. If:: appears, the route is originated locally.
Interface	The interface name. If “null” appears, the interface is originated locally.
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 BGP4+

45

In this chapter

Brocade's 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. 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
Brocade's 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.

This chapter describes the following:

- The address family configuration level for BGP4+.
- How to configure BGP4+.
- How to clear various BGP information, statistics, and counters.
- How to display BGP4+ information and statistics.

Address family configuration level

Brocade's implementation of BGP4+ includes a new configuration level: address family. For IPv6, *Brocade* currently supports the BGP4+ unicast address family configuration level only. (For IPv4, *Brocade* supports the BGP4 unicast and BGP4 multicast address family configuration levels.) The switch enters the BGP4+ unicast address family configuration level when you enter the following command while at the global BGP configuration level.

```
BigIron RX(config-bgp)# address-family ipv6 unicast
BigIron RX(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.

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.

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.

```
BigIron RX(config-bgp-ipv6u)# exit-address-family
BigIron RX(config-bgp)#
```

Entering this command returns you to the global BGP configuration level.

Configuring BGP4+

Before enabling BGP4+ on a switch, you must enable the forwarding of IPv6 traffic on the switch 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.

NOTE
You are required to configure a router_id when running IPv6 only routing protocols.

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.

```
BigIron RX(config)# router bgp
BGP: Please configure 'local-as' parameter in order to run BGP4.
BigIron RX(config-bgp)# local-as 1000
```

These commands enables the BGP4+ switch and configure the autonomous system (1000) in which your switch 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 switch you are configuring resides.

After enabling BGP4+, you can add neighbors to a BGP4+ switch by entering a commands such as the following.

```
BigIron RX(config-bgp)# address-family ipv6 unicast
BigIron RX(config-bgp-ipv6u)# neighbor 2001:4383:e0ff:783a::4 remote-as 1001
BigIron RX(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+. For more information about the parameters you can use with this command, refer to the *Router Configuration Guide*.

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 switch. You must repeat this procedure for each neighbor that you want to add to a local switch.

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 switch, enter the following commands.

```
BigIron RX(config-bgp)# address-family ipv6 unicast
BigIron RX(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 switch.
- Identify the neighbor interface over which the neighbor and local switch 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 switch, enter the following commands.

```
BigIron RX(config-bgp)# address-family ipv6 unicast
BigIron RX(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::/64. 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 switch will exchange prefixes, enter the following command.

```
BigIron RX(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::/64. 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 switch 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).

```
BigIron RX(config-bgp-ipv6u)# neighbor fe80:4398:ab30:45de::1 route-map out
next-hop
BigIron RX(config-bgp-ipv6u)# exit
BigIron RX(config)# route-map next-hop permit 10
BigIron RX(config-route-map)# match ipv6 address prefix-list next-hop-ipv6
BigIron RX(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 switch 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::/64. 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 switch. 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 switch 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 1121 and "Adding BGP4+ neighbors using link-local addresses" on page 1122.

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 do the following.

1. Create a peer group.
2. Add a neighbor to the local switch.
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:

```
BigIron RX(config-bgp)# address-family ipv6 unicast
BigIron RX(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 router

To add the IPv6 address 2001:efff:89::23 of a neighbor in remote AS 1001 to the BGP4+ neighbor table of a switch, enter the following command.

```
BigIron RX(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.

```
BigIron RX(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+ switch 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+ switch 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+ switch to advertise the default route, enter the following command.

```
BigIron RX(config-bgp-ipv6u)# default-information-originate
```

Syntax: [no] default-information-originate

You can also enable the BGP4+ switch 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+ switch to send the default route to a neighbor with the IPv6 address of 2001:efff:89::23, enter a command such as the following.

```
BigIron RX(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 switch 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.

```
BigIron RX(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 switch to redistribute routes from the following sources into BGP4+:

- Static IPv6 routes.
- Directly connected IPv6 networks.
- IPv6 IS-IS.
- OSPFv3.
- RIPng.

You can redistribute routes in the following ways:

- By route types, for example, the switch redistributes all IPv6 static and RIPng routes.
- By using a route map to filter which routes to redistribute, for example, the switch 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:

```
BigIron RX(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**, **isis**, **ospf**, **rip**, or **static**.

If you specify **isis** as the protocol, you can optionally specify the redistribution of level 1, level 1 and 2, or level 2 routes.

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 switch advertises individual BGP4+ routes for all the networks. The aggregation feature allows you to configure a switch to aggregate routes in a range of networks into a single IPv6 prefix. For example, without aggregation, a switch 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 switch 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.

```
BigIron RX(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 switch to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.

The **summary-only** keyword prevents the switch 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 switch to advertise the more specific routes in the specified route map.

The **attribute-map** <*map-name*> parameter configures the switch 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.

```
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# address-family ipv6 unicast
BigIron RX(config-bgp-ipv6u)# neighbor 2001:eff3:df78::67 remote-as 1001
BigIron RX(config-bgp-ipv6u)# neighbor 2001:eff3:df78::67 route-map in map1
BigIron RX(config-bgp-ipv6u)# exit
BigIron RX(config)# ipv6 prefix-list ipv6_uni seq 10 permit 2001:eff3::/32
BigIron RX(config)# route-map map1 permit 10
BigIron RX(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 switch 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:

```
BigIron RX# 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.

```
BigIron RX# 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 switch 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:

```
BigIron RX# 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.

```
BigIron RX# 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/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 1160.

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.

```
BigIron RX# clear ipv6 bgp neighbor 2000:e0ff:37::1 last-packet-with-error
BigIron RX# 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. For more information about cooperative filtering, refer to the "Configuring BGP4" chapter in the *Router Configuration Guide*.

To perform a hard reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
BigIron RX# 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 switch, the switch 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.

```
BigIron RX(config)# clear ipv6 bgp nei peer_group1 soft in prefix-filter
```

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 switch 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 switch and the neighbor clear all the routes they learned from each other. When the switch and neighbor establish a new BGP4+ session, they exchange route tables again. Use this method if you want the switch to relearn routes from the neighbor and resend its own route table to the neighbor.

If you use the **soft-outbound** keyword, the switch 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 switch 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 switch 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 switch 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 switch and all neighbors, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
BigIron RX# 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.

```
BigIron RX# 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 switch 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.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX(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. BGP4+ sends only the preferred route to the switch'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.

```
BigIron RX# 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 180 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 switch for reaching the route from the switch.
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 switch associates with routes from a specific neighbor. For example, if the switch receives routes to the same destination from two BGP4+ neighbors, the switch 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 switch 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 switch 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 switch in the same AS. • L – LOCAL. The route originated on this switch. **NOTE:** 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". 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 switch 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 switch.

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 switch does not have a valid RIPng, OSPFv3, IPv6 IS-IS, or static IPv6 route to the next hop.

To display details about BGP4+ routes, enter the following command at any level of the CLI.

```
BigIron RX# 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 181 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 180 on page 1134.
Status codes	For information about this field, refer to Table 180 on page 1134.
Prefix	For information about this field, refer to Table 180 on page 1134.
Status	For information about this field, refer to Table 180 on page 1134.
Age	The age of the advertised route, in seconds.
Next Hop	For information about this field, refer to Table 180 on page 1134.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. "Local router" indicates that the switch itself learned the route.
LOCAL_PREF	For information about this field, refer to Table 180 on page 1134.
MED	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.

TABLE 181 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 switch 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 switch. **NOTE:** 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". 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 180 on page 1134.
AS-PATH	For information about this field, refer to Table 180 on page 1134.
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 switch 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 switch.

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. For more information about regular expressions, refer to the “Configuring BGP4” chapter in the *Router Configuration Guide*.

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 switch does not have a valid RIPng, OSPFv3, IPv6 IS-IS, or static IPv6 route to the next hop.

Displaying BGP4+ route information

You can display all BGP4+ routes known by a switch, 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 switch, enter the following command at any level of the CLI.

```
BigIron RX# 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.

```
BigIron RX# 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.

```
BigIron RX# 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 182 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 switch.
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 switch for reaching the network from the switch.
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 switch associates with routes from a specific neighbor. For example, if the switch receives routes to the same destination from two BGP4+ neighbors, the switch 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 switch's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the switch typically has fewer route attribute entries than routes.

To display the IPv6 route-attribute entries table, enter the following command.

```
BigIron RX# 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 1157.

This display shows the following information:

TABLE 183 BGP4+ route-attribute entries information

This field...	Displays...
Total number of BGP Attribute Entries	The number of entries contained in the switch's BGP4+ route-attribute entries table.
Next Hop	The IPv6 address of the next hop switch 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 switch 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.

```
BigIron RX# 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.

```
BigIron RX# 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 184 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 switch 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 switch.

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.

```
BigIron RX# 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 switch's BGP policies filtered out. The switch 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 switch 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 185 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 switch for reaching the network from the switch.
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 185 Summary of filtered-out BGP4+ route information (Continued)

This field...	Displays...
Weight	The value that this switch associates with routes from a specific neighbor. For example, if the switch receives routes to the same destination from two BGP4+ neighbors, the switch 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 switch 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 switch 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 switch in the same AS. • L – LOCAL – The route originated on this switch. **NOTE:** 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". 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 switch, but the switch 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.

```
BigIron RX# 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 186 Detailed filtered-rut 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 185 on page 1144.
Status	For information about this field, refer to Table 185 on page 1144.

TABLE 186 Detailed filtered-rut BGP4+ route information (Continued)

This field...	Displays...
Age	The age of the route, in seconds.
Next hop	For information about this field, refer to Table 185 on page 1144.
Learned from peer	The IPv6 address of the neighbor from which this route is learned. “Local router” indicates that the switch itself learned the route.
Local pref	For information about this field, refer to Table 185 on page 1144.
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 switch 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 switch 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 switch in the same AS. • L – LOCAL – The route originated on this switch. • 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 switch, but the switch saved updates containing the filtered routes.
Weight	For information about this field, refer to Table 185 on page 1144.
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.

```
BigIron RX# 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 1159.

This display shows the following information:

TABLE 187 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the switch'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.

TABLE 187 Route flap dampening statistics

This field...	Displays...
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 1142.

Displaying BGP4+ neighbor information

You can display the following information about a switch'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.

```
BigIron RX# 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 switch's Transmission Control Block (TCB) for the TCP session between the switch 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 188 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/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 switch's session with the neighbor. The states are from the switch'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. **NOTE:** ACTIVE – BGP4+ is waiting for a TCP connection from the neighbor. 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 switch 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. **NOTE:** If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. 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 switch sends keep alive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the switch will wait for a KEEPALIVE or UPDATE message from a BGP4+ neighbor before deciding that the neighbor is dead.

TABLE 188 BGP4+ neighbor configuration information and statistics (Continued)

This field...	Displays...
RefreshCapability	Whether the switch has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
Messages Sent and Received	The number of messages this switch has sent to and received from the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
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

TABLE 188 BGP4+ neighbor configuration information and statistics (Continued)

This field...	Displays...
Last Connection Reset Reason (cont.)	• Reasons specific to the *Brocade* 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 switch 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 188 BGP4+ neighbor configuration information and statistics (Continued)

This field...	Displays...
Neighbor NLRI Negotiation	The state of the switch'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.
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 switch.
Local port	The TCP port the switch 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 switch.
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 switch 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 switch retransmitted because they were not acknowledged.

TABLE 188 BGP4+ neighbor configuration information and statistics (Continued)

This field...	Displays...
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.

Displaying routes advertised to a BGP4+ neighbor

You can display a summary or detailed information about the following:

- All routes a switch has advertised to a neighbor.
- A specified route a switch has advertised to a neighbor.

For example, to display a summary of all routes a switch has advertised to neighbor 2000:4::110, enter the following command at any level of the CLI.

```
BigIron RX# 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 189 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 switch for reaching the advertised route from the switch.
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 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 switch associates with routes from a specific neighbor. For example, if the switch receives routes to the same destination from two BGP4+ neighbors, the switch 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 switch received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • E – EBGP. The route was learned through a switch in another AS. • I – IBGP. The route was learned through a switch in the same AS. • L – LOCAL. The route originated on this switch.
AS-PATH	The AS-path information for the route.

For example, to display details about all routes a switch has advertised to neighbor 2000:4::110, enter the following command at any level of the CLI.

```
BigIron RX# 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:
```

This display shows the following information:

TABLE 190 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 189 on page 1156.
Status codes	For information about this field, refer to Table 189 on page 1156.
Prefix	For information about this field, refer to Table 189 on page 1156.
Status	For information about this field, refer to Table 189 on page 1156.
Age	The age of the advertised route, in seconds.
Next Hop	For information about this field, refer to Table 189 on page 1156.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the switch itself learned the route.
LOCAL_PREF	For information about this field, refer to Table 189 on page 1156.
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 189 on page 1156.
AS-PATH	The AS-path information for the route.
Adj RIB out count	The number of routes in the switch'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 switch's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the switch 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.

```
BigIron RX# 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 191 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 switch 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 switch 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.

TABLE 191 BGP4+ neighbor 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.
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.

```
BigIron RX# 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 192 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 1142.

Displaying last error packet from a BGP4+ neighbor

You can display information about the last packet that contained an error from any of a switch'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 switch's neighbors, enter the following command.

```
BigIron RX# 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 193 Last error packet information for BGP4+ neighbors

This field...	Displays...
Total number of BGP Neighbors	The total number of configured neighbors for a switch.
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.

```
BigIron RX# 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.

```
BigIron RX# 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 194 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 switch 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.

TABLE 194 Summary of route information received from a BGP4+ neighbor (Continued)

This field...	Displays...
Weight	The value that this switch associates with routes from a specific neighbor. For example, if the switch receives routes to the same destination from two BGP4+ neighbors, the switch 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 switch 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 switch 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 switch in the same AS. L – LOCAL. The route originated on this switch. 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 switch, but the switch 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.

```
BigIron RX# 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 195 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 194 on page 1161.
Status codes	For information about this field, refer to Table 194 on page 1161.
Prefix	For information about this field, refer to Table 194 on page 1161.
Status	For information about this field, refer to Table 194 on page 1161.
Age	The age of the route, in seconds.
Next hop	The next-hop switch for reaching the route from the switch.
Learned from peer	The IPv6 address of the neighbor from which this route is learned. “Local Router” indicates that the switch itself learned the route.
Local pref	For information about this field, refer to Table 194 on page 1161.
MED	The value of the route’s MED attribute. If the route does not have a metric, this field is blank.

TABLE 195 Detailed route information received from a BGP4+ neighbor (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 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 194 on page 1161.
AS Path	For information about this field, refer to Table 194 on page 1161.
Adj RIB out count	The number of routes in the switch'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 switch's current BGP4+ Routing Information Base (Adj-RIB-Out) for a specified neighbor.
- A specified route in a switch's current BGP4+ RIB for a specified neighbor.

The RIB contains the routes that the switch 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 switch's RIB for neighbor 2000:4::110, enter the following command at any level of the CLI.

```
BigIron RX# 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 196 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 switch for reaching the route from the switch.
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 switch associates with routes from a specific neighbor. For example, if the switch receives routes to the same destination from two BGP4+ neighbors, the switch 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 switch in another AS. • I – IBGP. The route was learned through a switch in the same AS. • L – LOCAL. The route originated on this switch.
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.

```
BigIron RX# 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:
```

This display shows the following information.

TABLE 197 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 196 on page 1165.
Status codes	For information about this field, refer to Table 196 on page 1165.
Prefix	For information about this field, refer to Table 196 on page 1165.
Status	For information about this field, refer to Table 196 on page 1165.
Age	The age of the RIB route, in seconds.
Next Hop	For information about this field, refer to Table 196 on page 1165.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. “Local Router” indicates that the switch itself learned the route.
LOCAL_PREF	For information about this field, refer to Table 196 on page 1165.
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 196 on page 1165.
AS-PATH	For information about this field, refer to Table 196 on page 1165.

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 switch’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.

```
BigIron RX# 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 198 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 switch for reaching the route from the switch.
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.

TABLE 198 Summary of best and unreachable routes from a BGP4+ neighbor (Continued)

This field...	Displays...
Weight	The value that this switch associates with routes from a specific neighbor. For example, if the switch receives routes to the same destination from two BGP4+ neighbors, the switch 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. • 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 switch 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 switch in the same AS. • L – LOCAL. The route originated on this switch. 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 switch, but the switch 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.

```
BigIron RX# 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 199 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 198 on page 1167.
Status codes	For information about this field, refer to Table 198 on page 1167.
Prefix	For information about this field, refer to Table 198 on page 1167.
Status	For information about this field, refer to Table 198 on page 1167.
Age	The age of the route, in seconds.
Next Hop	For information about this field, refer to Table 198 on page 1167.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the switch itself learned the route.
LOCAL_PREF	For information about this field, refer to Table 198 on page 1167.
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 198 on page 1167.
AS-PATH	For information about this field, refer to Table 198 on page 1167.

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.

```
BigIron RX# 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 200 BGP4+ neighbor route summary information

This field...	Displays...
IP Address	The IPv6 address of the neighbor
Routes Received	How many routes the switch has received from the neighbor during the current BGP4+ session. • Accepted/Installed – Indicates how many of the received routes the switch accepted and installed in the BGP4+ route table. • Filtered/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 switch 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 switch received better routes from other sources (such as OSPFv3, RIPng, IPv6 IS-IS, or static IPv6 routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the switch 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.

TABLE 200 BGP4+ neighbor route summary information (Continued)

This field...	Displays...
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 switch has received. • Replacements – The number of replacement routes the switch has received.
NLRIs Discarded due to	Indicates the number of times the switch discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The switch'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 switch ID. • Cluster_ID – The cluster list contained the local cluster ID, or contained the local switch ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the switch has advertised to this neighbor. • To be Sent – The number of routes the switch has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the switch has queued up to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the switch has sent to this neighbor in UPDATE messages. • Withdraws – The number of routes the switch has sent to the neighbor to withdraw. • Replacements – The number of routes the switch has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the switch 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 switch.

For example, to display configuration information for a peer group named peer1, enter the following command at any level of the CLI.

```
BigIron RX# 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 switch, enter the following command at any level of the CLI.

```
BigIron RX# 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 201 BGP4+ summary information

This field...	Displays...
Router ID	The switch's router ID.
Local AS Number	The BGP4+ AS number in which the switch resides.
Confederation Identifier	The AS number of the confederation in which the switch resides.

TABLE 201 BGP4+ summary information (Continued)

This field...	Displays...
Confederation Peers	The numbers of the local ASs contained in the confederation. This list matches the confederation peer list you configure on the switch.
Maximum Number of Paths Supported for Load Sharing	The maximum number of route paths across which the switch 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. For more information, refer to the "Configuring BGP4" chapter in the *Router Configuration Guide*.
Number of Neighbors Configured	The number of BGP4+ neighbors configured on this switch.
Number of Routes Installed	The number of BGP4+ routes in the switch's BGP4+ route table. To display the BGP4+ route table, refer to "Displaying the BGP4+ route table" on page 1133.
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 switch's route-attributes table. To display the route-attribute table, refer to "Displaying BGP4+ route-attribute entries" on page 1140.
Neighbor Address	The IPv6 addresses of this switch's BGP4+ neighbors.
AS#	The AS number.

TABLE 201 BGP4+ summary information (Continued)

This field...	Displays...
State	The state of this switch's neighbor session with each neighbor. The states are from this switch's perspective of the session, not the neighbor's perspective. The state values can be one of the following for each switch: • 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. **NOTE:** ACTIVE – BGP4+ is waiting for a TCP connection from the neighbor. 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 switch 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 switch installed in the BGP4+ route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this switch 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 switch has sent to the neighbor.
ToSend	The number of routes the switch has queued to send to this neighbor.

Chapter

Configuring IPv6 MBGP

46

In this chapter

Brocade's 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. 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.

Configuration considerations

The following are the configuration considerations:

- IPv6 MBGP does not redistribute DVMRP routes. It redistributes static routes only.
- You cannot redistribute IPv6 MBGP routes into BGP4.
- The device supports 8192 multicast routes by default. You may need to increase the maximum number of multicast routes for MBGP. You can configure the device to support up to 153,600 multicast routes.

Configuring IPv6 MBGP

1. Optional – Set the maximum number of multicast routes supported by the device.
2. Enable IPv6 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 BigIron RX in order for the device to send multicast prefixes to other multicast routers.
   - 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 IPv6 MBGP routers.
4. Optional – Configure an IPv6 MBGP default route.
5. Optional – Configure an IPv6 multicast static route.

6. Optional – Configure an IPv6 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

The BigIron RX supports up 1024 – 153,600 multicast routes.

NOTE
This procedure requires a software reload to place the change into effect.

To increase the maximum number of multicast routes supported on the device, enter commands such as the following.

```
BigIron RX(config)# system-max multicast-route 12000
BigIron RX(config)# write memory
BigIron RX(config)# end
BigIron RX# 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 and can be from 1024 – 153,600.

Enabling IPv6 MBGP

To enable IPv6 MBGP, you must enable PIM SM or DM and IPv6 BGP. Enter commands such as the following.

```
BigIron RX> enable
BigIron RX# configure terminal
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# ipv6 address 3001::1
BigIron RX(config-if-1/1)# ipv6 pim
BigIron RX(config-if-1/1)# exit
BigIron RX(config)# router bgp
ipv6BGP: Please configure 'local-as' parameter in order to enable ipv6BGP.
BigIron RX(config-bgp)# local-as 10
```

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 IPv6 multicast address family. Enter the following command to enter the IPv6 multicast address family level.

```
BigIron RX(config-bgp)#address-family ipv6 multicast
BigIron RX(config-bgp-ipv6m)#
```

Syntax: address-family ipv6 multicast l unicast

Adding IPv6 MBGP neighbors

To add an MBGP neighbor, enter a command such as the following.

```
BigIron RX(config-bgp-ipv6m)# neighbor 3001::1 remote-as 44
```

This command adds a router with IPv6 address 3001::1 as an MBGP neighbor.

The **remote-as 44** parameter specifies that the neighbor is in remote BGPv6 AS 44. The BigIron RX will exchange only multicast routes with the neighbor.

NOTE
If the BigIron RX 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 IPv6 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 IPv6 multicast address family level. Here is the full syntax for the neighbor command.

Syntax: [no] neighbor <*ipv6-addr*> | <*peer-group-name*>
[advertisement-interval <*num*>]
[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]
[timers keep-alive <*num*> hold-time <*num*>]
[update-source loopback <*num*>]
[weight <*num*>*]*

The <*ipv6-addr*> | <*peer-group-name*> parameter indicates whether you are configuring an individual neighbor or a peer group. If you specify a neighbor's IPv6 address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group.

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.

Optional configuration tasks

The following sections describe how to perform some optional BGPv6 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. For information on these other tasks, refer to Chapter 26, "Configuring BGP4 (IPv4 and IPv6)".

Advertising routes from the local AS to MBGP

You can configure the BigIron RX 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 IPv6 multicast routes. The corresponding IPv6 route must be present in the IPv6 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 BigIron RX advertises MBGP routes only for the networks you identify using the **network** command or that are redistributed into MBGP from IPv6 multicast route tables.

NOTE
The exact route must exist in the IPv6 multicast route table so that the BigIron RX can create a local MBGP route.

To configure the BigIron RX to advertise network 207.95.22.0/24 as a multicast route, enter the following command.

```
BigIron RX(config-bgp-ipv6m)# network 207.95.22.0 255.255.255.0
```

Syntax: network <*ipv6-addr*> <*ipv6-mask*> [route-map <*map-name*>] [backdoor] [weight <*num*>]

The <*ipv6-addr*> is the network number and the <*ipv6-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.

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. Here is an example.

```
BigIron RX(config)# access-list 10 permit 2001:100::/32
BigIron RX(config)# route-map mbgpmap permit 1
BigIron RX(config-routemap mbgpmap)# match ipv6 address 10
BigIron RX(config-routemap mbgpmap)# exit
BigIron RX(config)# router bgp
BigIron RX(config-bgp-ipv6m)# redistribute connected route-map mbgpmap
```

The first command configures an ipv6 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 ipv6 ACL 10. The BigIron RX 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.

Configuring static IPv6 multicast routes

To configure static IPv6 multicast routes, enter a command such as the following.

```
BigIron RX(config)# ipv6 mroute 8eff::0/32 4fee:2343:0:ee44::1
```

If you configure more than one static multicast route, the BigIron RX Series router 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.

Syntax: [no] ipv6 mroute <*ipv6-addr*> <*ipv6-mask*> [<*next-hop-ipv6-addr*> | ethernet <*slot/port*> | ve <*num*> | null0] [<*cost*>] [distance < *num*>]

The **ipv6-addr** and **ipv6-mask** parameters specifies the PIM source for the route.

The **ethernet** <*slot/port*> parameter specifies a physical port.

The **ve** <*num*> parameter specifies a virtual interface.

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

Regardless of the administrative distances, the BigIron RX Series router always prefers directly connected routes over other routes.

Aggregating routes advertised to IPv6 BGP neighbors

By default, the BigIron RX advertises individual MBGP routes for all the multicast networks. The aggregation feature allows you to configure the BigIron RX to aggregate routes in a range of networks into a single CIDR number.

To aggregate MBGP routes, enter the following command.

```
BigIron RX(config-bgp-router)# aggregate-address 8eff::0/32 4fee:2343:0:ee44::1
```

Syntax: aggregate-address <*ipv6-addr*> <*ipv6-mask*> [as-set] [summary-only] [suppress-map <*map-name*>] [advertise-map <*map-name*>] [attribute-map <*map-name*>]

The <*ipv6-addr*> and <*ipv6-mask*> parameters specify the aggregate value for the networks.

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 BigIron RX to advertise the more specific routes in the specified route map.

The **attribute-map** <*map-name*> parameter configures the BigIron RX 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 IPv6 MBGP information

All of the IPv6 BGP show commands have IPv6 MBGP equivalents. Use **ipv6 mbgp** instead of **ipv6 bgp** in the command syntax. For example, to display the MBGP route table, enter the **show ipv6 mbgp routes** command instead of the **show ipv6 bgp routes** command. Table 202 lists the MBGP show commands and describes their output. For information about a command, refer to Chapter 26, “Configuring BGP4 (IPv4 and IPv6)”.

TABLE 202 IPv6 MBGP Show commands

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.

TABLE 202 IPv6 MBGP Show commands (Continued)

Command	Description
show ipv6 mbgp peer-group	Displays information about IPv6 MBGP peer groups.
show ipv6 mbgp routes	Displays IPv6 MBGP routes.
show ipv6 mbgp *<ipv6-addr>[/<prefix>]*	Displays a specific IPv6 MBGP route.
show ipv6 mbgp attribute-entries	Displays IPv6 MBGP route attributes.
show ipv6 mbgp dampened-paths	Displays IPv6 MBGP paths that have been dampened by route flap dampening.
show ipv6 mbgp flap-statistics	Displays route flap dampening statistics.
show ipv6 mbgp filtered-routes	Displays routes that have been filtered out.

Displaying summary MBGP information

To display summary MBGP information, enter the following command at any CLI prompt.

```
BigIron RX# show ipv6 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 ipv6 mbgp summary

NOTE
This command's display looks similar to the display for the **show ipv6 bgp config** command. However, the **show ipv6 mbgp config** command lists only the MBGP neighbors, whereas the **show ipv6 bgp config** command lists only the BGP neighbors.

Displaying the Active MBGP Configuration

To display the active MBGP configuration information contained in the running-config without displaying the entire running-config, enter the following command at any level of the CLI.

```
BigIron RX# show ipv6 mbgp config
Current BGP configuration:

router bgp
 local-as 200
 neighbor 166.1.1.2 remote-as 200

 address-family ipv6 unicast
 no neighbor 166.1.1.2 activate
 exit-address-family

 address-family ipv6 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 ipv6 mbgp config

NOTE
This command displays exactly the same information as the **show ipv6 bgp config** command. Each command displays both the BGP and MBGP configuration commands that are in the running-config.

Displaying MBGP neighbors

To view MBGP neighbor information including the values for all the configured parameters, enter the following command. This display is similar to the **show ipv6 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.

```
BigIron RX # show ipv6 mbgp neighbor 4fee:2343:0:ee44::1
    Total number of BGP Neighbors: 1
1   ipv6 Address: 8eff::0/32, 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 ipv6  multicast  capability
      Peer configured for  ipv6 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
```

This example shows how to display information for a specific neighbor, by specifying the neighbor's IPv6 address with the command. 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 Neighbor NLRI Negotiation section (shown in bold type) lists the types of routes that this BigIron RX 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 BigIron RX's Transmission Control Block (TCB) for the TCP session between the BigIron RX and its neighbor. These fields are described in detail in section 3.2 of RFC 793, "Transmission Control Protocol Functional Specification".

Syntax: show ipv6 mbgp neighbors [<*ipv6-addr*>]

The <*ipv6-addr*> parameter specifies the neighbor's IPv6 address.

Displaying MBGP routes

To display the MBGP route table, enter the following command.

```
BigIron RX#show ipv6 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 ipv6 mbgp routes

Displaying the IPv6 multicast route table

To display the IPv6 multicast route table, enter the following command.

```
BigIron RX#show ipv6 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 ipv6 mroute [<*ipv6-addr*> <*ipv6-mask*> | bgp | static]

The **<ipv6-addr> <ipv6-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.

Chapter

IPv6 Access Control Lists (ACLs) 47

In this chapter

IPv6 ACLs

Brocade supports IPv6 Access Control Lists (ACLs), which you can use for traffic filtering. You can configure up to 100 IPv6 ACLs.

An IPv6 ACL is composed of one or more conditional statements that pose 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 in each IPv6 ACL 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 IPv6 packets on specified interfaces. You can apply only one IPv6 ACL to an interface's incoming traffic. When an interface sends or receives an IPv6 packet, it applies the statements within the ACL in their order of appearance to the packet. As soon as a match occurs, the *BigIron RX* takes the specified action (permit or deny the packet) and stops further comparison for that packet.

NOTE
IPv6 ACLs are supported on inbound traffic and are implemented in hardware, making it possible for the *BigIron RX* to filter traffic at line-rate speed on 10 Gigabit interfaces.

Brocade's IPv6 ACLs enable traffic filtering based on the following information:

- IPv6 protocol
- Source IPv6 address
- Destination IPv6 address
- IPv6 message type
- 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 website by denying all TCP port 80 (HTTP) packets from a specified source IPv6 address to the website's IPv6 address.

IPv6 ACLs also provide support for filtering packets based on DSCP.

This chapter contains the following sections:

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, 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.

Configuring an IPv6 ACL

To configure an IPv6 ACL, you must do the following:

- Create the ACL
- Apply the ACL to an interface

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.

```
BigIron RX(config)# ipv6 access-list fdry
BigIron RX(config-ipv6-access-list-fdry)# deny tcp host 2000:2382:e0bb::2 any eq
telnet
BigIron RX(config-ipv6-access-list-fdry)# permit ipv6 any any
BigIron RX(config-ipv6-access-list-fdry)# exit
BigIron RX(config)# int eth 1/1
BigIron RX(config-if-1/1)# ipv6 traffic-filter fdry in
BigIron RX(config)# write memory
```

Here is another example of commands for configuring an ACL and applying it to an interface.

```
BigIron RX(config)# ipv6 access-list netw
BigIron RX(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
BigIron RX(config-ipv6-access-list-netw)# deny ipv6 host 2000:2383:e0ac::2 host
2000:2383:e0aa:0::24
BigIron RX(config-ipv6-access-list-netw)# deny udp any any
BigIron RX(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 would deny all incoming IPv6 traffic on the ports to which you assigned the ACL.

The following commands apply the ACL "netw" to the incoming traffic on port 1/2 and to the incoming traffic on port 4/3.

```
BigIron RX(config)# int eth 1/2
BigIron RX(config-if-1/2)# ipv6 traffic-filter netw in
BigIron RX(config-if-1/2)# exit
BigIron RX(config)# int eth 4/3
BigIron RX(config-if-4/3)# ipv6 traffic-filter netw in
BigIron RX(config)# write memory
```

Here is another example of an ACL.

```
BigIron RX(config)# ipv6 access-list nextone
BigIron RX(config-ipv6-access-list rtr)# deny tcp 2001:1570:21::/24
2001:1570:22::/24
BigIron RX(config-ipv6-access-list rtr)# deny udp any range 5 6 2001:1570:22::/24
BigIron RX(config-ipv6-access-list rtr)# permit ipv6 any any
BigIron RX(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 next condition denies UDP packets from any source with source UDP port in ranges 5 to 6 and whose destination is to the 2001:1570:22::/24 network.

The third condition permits all packets containing source and destination addresses that are not explicitly denied by the first two. Without this entry, the ACL would deny all incoming IPv6 traffic on the ports to which you assign the ACL.

A **show running-config** command displays the following.

```
 BigIron RX(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 displays the following.

```
 BigIron RX(config)# sh 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.

```
BigIron RX(config)# int eth 2/1
BigIron RX(config-if-2/1)# ipv6 traffic-filter rtr in
BigIron RX(config-if-2/1)# exit
BigIron RX(config)# int eth 2/2
BigIron RX(config-if-2/2)# ipv6 traffic-filter rtr in
BigIron RX(config)# write memory
```

Default and implicit IPv6 ACL action

The default action when no IPv6 ACLs are configured on an interface is to permit all IPv6 traffic. However, 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.

- 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 permit entry permits packets that are not denied by the deny entries.

Every IPv6 ACL has the following implicit conditions as its last match conditions.

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 access-list if you want to permit IPv6 traffic that were not denied by the previous statements.

The conditions are applied in the order shown above, with deny ipv6 any any as the last condition applied.

For example, if you want to deny ICMP neighbor discovery acknowledgement, then permit any remaining IPv6 traffic, enter commands such as the following.

```
BigIron RX(config)# ipv6 access-list netw
BigIron RX(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
BigIron RX(config-ipv6-access-list-netw)# deny icmp any any nd-na
BigIron RX(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 entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL will deny all incoming 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 the example below.

```
BigIron RX(config)# ipv6 access-list netw
BigIron RX(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
BigIron RX(config-ipv6-access-list-netw)# permit icmp any any nd-na
BigIron RX(config-ipv6-access-list-netw)# permit icmp any any nd-ns
BigIron RX(config-ipv6-access-list-netw)# deny icmp any any
BigIron RX(config-ipv6-access-list-netw)# permit ipv6 any any
```

ACL syntax

NOTES: The following features are not supported:

- ipv6-operator flow-label
- **ipv6-operator fragments** when any protocol is specified. The option "fragments" can be specified only when "permit/deny ipv6" is specified. If you specify "tcp" or any other protocol instead of "ipv6" the keyword, "fragments" cannot be used.
- **ipv6-operator routing** when any protocol is specified. (Same limitation as for **ipv6-operator fragments**)

When creating ACLs, use the appropriate syntax below for the protocol you are filtering.

For IPv6 and supported protocols other than ICMP, TCP, or UDP

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*>]]
[802.1p-priority-matching <*number*>]
[dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>] | [dscp-marking <*dscp-value*> dscp-cos-mapping] | [dscp-cos-mapping]

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*>]
[802.1p-priority-matching <*number*>]
[dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>]
[dscp-marking <*dscp-value*> dscp-cos-mapping]
[dscp-cos-mapping]

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*>]]
[match-all <*tcp flags*>] | [match-any <*tcp flags*>] |established
[802.1p-priority-matching <*number*>]
[dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>]
[dscp-marking <*dscp-value*> dscp-cos-mapping]
[dscp-cos-mapping]

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*>]]
[802.1p-priority-matching <*number*>]
[dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>]
[dscp-marking <*dscp-value*> dscp-cos-mapping]
[dscp-cos-mapping]

TABLE 203 Syntax descriptions

Arguments...	Description...
ipv6 access-list *<acl name>*	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.
permit	The ACL will permit (forward) packets that match a policy in the access list.
deny	The ACL will deny (drop) packets that match a policy in the access list.
icmp	Indicates the you are filtering ICMP packets.
protocol	The type of IPv6 packet you are filtering. You can specify a well-known name for some protocols 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. IPv6 protocols include: • **AHP** – Authentication Header • **ESP** – Encapsulating Security Payload • **IPv6** – Internet Protocol version 6 • **SCTP** – Stream Control Transmission Protocol
<ipv6-source-prefix>/<prefix-length>	The *<ipv6-source-prefix>/<prefix-length>* parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the *<ipv6-source-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.
<ipv6-destination-prefix>/<prefix-length>	The *<ipv6-destination-prefix>/<prefix-length>* parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the *<ipv6-destination-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
any	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.
host	Allows 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.
icmp-type	ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.
icmp code	ICMP packets, which are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255,
icmp-message	ICMP packets,are filtered by ICMP messages. Refer to "ICMP message configurations" on page 1194 for a list of ICMP message types.
tcp	Indicates the you are filtering TCP packets.
udp	Indicates the you are filtering UDP packets.

TABLE 203 Syntax descriptions (Continued)

Arguments...	Description...
<ipv6-source-prefix>/<prefix-length>	The *<ipv6-source-prefix>/<prefix-length>* parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the *<ipv6-source-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.
<ipv6-destination-prefix>/<prefix-length>	The *<ipv6-destination-prefix>/<prefix-length>* parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the *<ipv6-destination-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
any	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.
host	Allows 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.
tcp-udp-operator	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 is the number of the port.

TABLE 203 Syntax descriptions (Continued)

Arguments...	Description...
ipv6-operator	Allows you to filter the packets further by using one of the following 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 not applicable to filtering based on source or destination port, TCP flags, and ICMP flags. • **routing** – The policy applies only to IPv6 source-routed packets. **NOTE:** This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags. • **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. You can specify which flags inside the TCP header need to be matched. Specify any of the following flags for <*tcp- flags*>: • + | – urg = Urgent • + | – ack= Acknowledge • + | – psh + Push • + | – rst = Reset • + | – syn = Synchronize • + | – fin = Finish
match-all <*tcp- flags*> **match-any** <*tcp-flag*>	Enter **match-all** <*tcp- flags*> if you want all the flags you specify to be matched from a TCP session. Use **match-any** <*tcp-flag*> if any of the flags will be matched. You can enter more than one TCP flag. Separate each flag with a space, using a + or – to indicate if the matching condition requires the bit to be set to 1 (+) or 0 (–).
802.1p-priority-matching <*number*>	If you want to match only those packets that have the same 802.1p priorities as specified in the ACL. Enter 0 – 7.
dscp-marking <*number*>	Use the **dscp-marking** <*number*> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the DSCP value that you specify to the packet. Enter 0 – 63.
802.1p-priority-marking <*number*>	Use the **802.1p-priority-marking** <*number*> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the 802.1p priority that you specify to the packet. Enter 0 – 7.
internal-priority-marking <*number*>	Use the **internal-priority-marking** <*number*> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the internal priority that you specify to the packet. Enter 0 – 7.

TABLE 203 Syntax descriptions (Continued)

Arguments...	Description...
dscp-marking <*number*>	Use the **dscp-marking** <*number*> **dscp-cos-mapping** parameters parameters to specify a DSCP value and map that value to an internal QoS table to obtain the packet's new QoS value. The following occurs when you use these parameters. • You enter 0 – 63 for the **dscp-marking** <*number*> parameter. • The **dscp-cos-mapping** parameter takes the DSCP value you specified and compares it to an internal QoS table, which is indexed by DSCP values. The corresponding 802.1p priority, internal forwarding priority, and DSCP value is assigned to the packet.
dscp-cos-mapping	Use **dscp-cos-mapping** if you want to use the DSCP value in the packet's header to alter its QoS value. When you enter **dscp-cos-mapping**, the DSCP value in the packet's header is compared to a column in the internal QoS table. The 802.1p priority, internal forwarding priority, and DSCP value that are mapped to the matching column is assigned to the packet.

ICMP message configurations

If you want to specify an ICMP message, you can enter one of the following:

- 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

- sequence
- time-exceeded
- unreachable

NOTE
If you do not specify a message type, the ACL applies to all types ICMP messages types.

Applying an IPv6 ACL to an interface

To apply an IPv6 ACL, for example “access1”, to an interface, enter commands such as the following.

```
BigIron RX(config)# interface ethernet 3/1
BigIron RX(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.

Syntax: ipv6 traffic-filter <*ipv6-acl-name*> in

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 interface.

Adding TCP flags to an IPv6 ACL entry

You can add aTCP flags to entries in an IPv6 ACL. The TCP flag will appear in the output of **show** commands that display ACL information.

Enter **match-all** <*tcp- flags*> if you want all the flags you specify to be matched from a TCP session. Use **match-any** <*tcp-flag*> if any of the flags will be matched. You can enter more than one TCP flag. Separate each flag with a space, using a + or – to indicate if the matching condition requires the bit to be set to 1 (+) or 0 (–).

This example applies the TCP flags to allow TCP packets only if the TCP flags SYN and ACK are set and the FIN flag is not set.

```
BigIron RX> enable
BigIron RX# configure terminal
BigIron RX(config)# ipv6 access-list fdry
BigIron RX (config-ipv6-access-list fdry)# permit tcp any any match-all +ack +syn
BigIron RX(config-ipv6-access-list fdry)# permit tcp any any match-any -urg +syn
-psh
BigIron RX (config-ipv6-access-list fdryl)# end
```

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.

```
BigIron RX(config)#ipv6 access-list rtr
BigIron RX(config-ipv6-access-list rtr)# remark This entry permits ipv6 packets
from 3002::2 to any destination
BigIron RX(config-ipv6-access-list rtr)# permit ipv6 host 3000::2 any
BigIron RX(config-ipv6-access-list rtr)# remark This entry denies udp packets from
any source to any destination
BigIron RX(config-ipv6-access-list rtr)# deny udp any any
BigIron RX(config-ipv6-access-list rtr)# remark This entry denies IPv6 packets
from any source to any destination
BigIron RX(config-ipv6-access-list rtr)# deny ipv6 any any
BigIron RX(config-ipv6-access-list rtr)# write memory
```

Syntax: 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.

```
BigIron RX(config)# ipv6 access-list netw
BigIron RX(config-ipv6-access-list netw) remark-entry sequence 10 This entry
permits ipv6 packets from 3000::2 to any destination
BigIron RX(config-ipv6-access-list netw)# remark-entry sequence 20 This entry
denies UDP packets from any source to any destination
BigIron RX(config-ipv6-access-list netw)# remark-entry sequence 30 This entry
denies IPv6 packets from any source to any destination
```

Syntax: 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.

```
BigIron RX# 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.

```
BigIron RX# 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.

Displaying ACLs

To display the ACLs configured on a device, enter the **show ipv6 access-list** command. Here is an example.:

```
BigIron RX# show ipv6 access-list
ipv6 access-list fdry: 8 entries
 10: permit ipv6 host 3000::2 any
 20: permit udp 3000::/16 any gt nfs
 30: deny icmp host 5000::5 host 6000::3 echo-request
 40: permit ipv6 host 3002::2 any
 50: deny udp 3000::/16 4000::/16 gt nfs
 60: permit tcp any any established
 70: permit udp any any gt nfs
 80: remark this is last entryipv6 access-list fdry: 3 entries
```

Syntax: show ipv6 access-list [<*access-list-name*>*]*

Chapter

48

Configuring OSPF Version 3

In this chapter

OSPF version 3

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 switch 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.

This chapter describes the following:

- The differences between OSPF versions 2 and 3
- The link state advertisement types for OSPF version 3
- How to configure OSPF version 3
- How to display OSPF version 3 information and statistics

IPv6 supports OSPF version 3 (OSPFv3), which functions similarly to OSPF version 2, the current version that IPv4 supports, except for the following enhancements:

- Support for IPv6 addresses and prefixes.
- In general, you can configure several IPv6 addresses on a router interface. OSPFv3 imports all or none of the address prefixes configured on a router interface. You cannot select which addresses to import.
- You can run one instance of OSPF version 2 and one instance of OSPFv3 concurrently on a link.
- IPv6 link state advertisements (LSAs).

In addition, Brocade implements some new commands that are specific to OSPFv3. This section describes the commands that are specific to OSPFv3.

NOTE
Although OSPF versions 2 and 3 function similarly to each other, Brocade has implemented the user interface for each version independently of each other. Therefore, any configuration of OSPF version 2 features will not affect the configuration of OSPFv3 features and vice versa.

NOTE
You are required to configure a router ID when running only IPv6 routing protocols.

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 do the following:

- 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 ABR without a physical connection to a backbone area and the Brocade 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 Brocade 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 Brocade device to run OSPFv3, you must do the following:

- Enable the forwarding of IPv6 traffic on the Brocade 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 Chapter 43, "Configuring Basic IPv6 Connectivity".

By default, OSPFv3 is disabled. To enable OSPFv3, you must enable it globally.

To enable OSPFv3 globally, enter the following command.

```
BigIron RX(config-ospf-router)#ipv6 router ospf
BigIron RX(config-ospf6-router)#
```

After you enter this command, the Brocade device enters the IPv6 OSPF configuration level, where you can access several commands that allow you to configure OSPFv3.

Syntax: [no] ipv6 router ospf

To disable OSPFv3, enter the **no** form of this command. If you disable OSPFv3, the Brocade device removes all the configuration information for the disabled protocol from the running-config. 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.

The CLI displays a warning message such as the following.

```
BigIron RX(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 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-config 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-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's Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) 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.

```
BigIron RX(config-ospf6-router)# area 0.0.0.0
BigIron RX(config-ospf6-router)# area 200.5.0.0
BigIron RX(config-ospf6-router)# area 192.5.1.0
BigIron RX(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 – 2,147,483,647.

NOTE
You can assign one area on a router interface.

Assigning a totally stubby area

By default, the Brocade device sends summary LSAs (LSA type 3) into stub areas. You can further reduce the number of LSAs sent into a stub area by configuring the Brocade 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 Brocade device still accepts summary LSAs from OSPF neighbors and floods them to other areas. The Brocade 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 Brocade 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.

```
BigIron RX(config-ospf6-router)# area 40 stub 99 no-summary
```

Syntax: 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 – 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 – 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.

```
BigIron RX(config)# interface Ethernet 3/1
BigIron RX(config-if-e100-3/1)# ipv6 ospf area 192.5.0.0
```

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 – 2,147,483,647.

To remove the interface from the specified area, use the **no** form of this command.

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.
- When assigned from the router interface requiring a logical connection, the neighbor router field is the router ID (IPv4 address) of the router that is physically connected to the backbone. When assigned from the router interface with the physical connection, the neighbor router is the router ID (IPv4) address of the router requiring a logical connection to the backbone.

NOTE
By default, the Brocade router ID is the IPv4 address configured on the lowest numbered loopback interface. If the Brocade device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.

NOTE
When you establish an area virtual link, you must configure it on both of the routers (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.

```
BigIron RX(config-ospf6-router)# area 1 virtual-link 209.157.22.1
```

To define the virtual link on ABR2, enter the following command on ABR2.

```
BigIron RX(config-ospf6-router)# area 1 virtual-link 10.0.0.1
```

Syntax: area <*number*> | <*ipv4-address*> virtual-link <*router-id*>

The **area** <*number*> | <*ipv4-address*> parameter specifies the transit area.

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 need to communicate with one another, the source address included in the packets must be a global IPv6 address. Therefore, you must determine the global IPv6 address to be used by the routers for communication across the virtual link. You can specify that a router uses the IPv6 global address assigned to one of its interfaces.

For example, to specify the global IPv6 address assigned to Ethernet interface 3/1 on ABR1 as the source address for the virtual link on ABR1, enter the following command on ABR1.

```
BigIron RX(config-ospf6-router)# virtual-link-if-address interface ethernet 3/1
```

To specify the global IPv6 address assigned to tunnel interface 1 on ABR2 as the source address for the virtual link on ABR2, enter the following command on ABR2.

```
BigIron RX(config-ospf6-router)# virtual-link-if-address interface tunnel 1
```

Syntax: virtual-link-if-address interface ethernet <*port*> | loopback <*number*> | tunnel <*number*> | ve <*number*>

The **ethernet | loopback | tunnel | ve** parameter specifies the interface from which the router derives the source IPv6 address for communication across the virtual link. 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 respective interface.

To delete the source address for the virtual link, use the **no** form of this command.

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 1 – 65535 seconds. The default is 40 seconds.
- **Hello-interval** - The length of time between the transmission of hello packets. The range is 1 – 65535 seconds. The default is 10 seconds.
- **Retransmit-interval** - 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** - 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.

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 remember to 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 to 60 seconds on the virtual links defined on ABR1 and ABR2, enter the following command on ABR1.

```
BigIron RX(config-ospf6-router)# area 1 virtual-link 209.157.22.1
dead-interval 60
```

Enter the following command on ABR2.

```
BigIron RX(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.

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 discussed 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 Brocade device advertises its interfaces and their costs to OSPFv3 neighbors. For example, if an interface has an OSPF cost of ten, the Brocade 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 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'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
- 155 Mbps port's cost = 100/155 = 0.65, which is rounded up to 1
- 622 Mbps port's cost = 100/622 = 0.16, which is rounded up to 1
- 2488 Mbps port's cost = 100/2488 = 0.04, which is rounded up to 1

The bandwidth for interfaces that consist of more than one physical port is calculated as follows:

- Trunk 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 – 4294967 Mbps.

If a change to the reference bandwidth results in a cost change to an interface, the Brocade device sends a link state update to update the costs of interfaces advertised by the Brocade device.

NOTE
If you specify the cost for an individual interface, the cost you specify overrides the cost calculated by the software.

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 0.
- 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 not affected by the auto-cost feature.

For example, to change the reference bandwidth to 500, enter the following command.

```
BigIron RX(config-ospf6-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
- 155 Mbps port's cost = 500/155 = 3.23, which is rounded up to 4
- 622 Mbps port's cost = 500/622 = 0.80, which is rounded up to 1
- 2488 Mbps port's 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 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 **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 Brocade 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 Brocade device redistributes all IPv6 static and RIPng routes.
- By using a route map to filter which routes to redistribute, for example, the Brocade 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.

```
BigIron RX(config-ospf6-router)# redistribute static
BigIron RX(config-ospf6-router)# redistribute rip
BigIron RX(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 Brocade 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 1208

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 Brocade 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 default metric for routes redistributed into OSPF version 3" on page 1208

For example, to configure a route map and use it for redistribution of routes into OSPFv3, enter commands such as the following.

```
BigIron RX(config)# ipv6 route 2001:1::/32 4823:eoff:343e::23
BigIron RX(config)# ipv6 route 2001:2::/32 4823:eoff:343e::23
BigIron RX(config)# ipv6 route 2001:3::/32 4823:eoff:343e::23 metric 5
BigIron RX(config)# route-map abc permit 1
BigIron RX(config-routemap abc)# match metric 5
BigIron RX(config-routemap abc)# set metric 8
BigIron RX(config-routemap abc)# ipv6 router ospf
BigIron RX(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 0.

If the **metric** parameter for the **redistribute** command is not set and the **default-metric** command is set to 0, 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 1206.

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 1217 and "Changing the reference bandwidth for the cost on OSPFv3 interfaces" on page 1205.

To assign a default metric of 4 to all routes imported into OSPFv3, enter the following command.

```
BigIron RX(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 Brocade 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 1206).

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.

```
BigIron RX(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 Brocade 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 Brocade 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 Brocade 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 Brocade 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 Brocade 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.

```
BigIron RX(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.

```
BigIron RX# 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.

```
BigIron RX(config)# ipv6 prefix-list  filterOspfRoutes seq 5 deny 3010::/64
BigIron RX(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.

```
BigIron RX(config)# ipv6 router ospf
BigIron RX(config-ospf6-router)# distribute-list prefix-list filterOspfRoutes in
```

Syntax: [no] distribute-list prefix-list <*name*> in [<*interface*>]

After this distribution list is configured, route 3010::/64 would be omitted from the OSPFv3 route table.

```
BigIron RX# 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.

```
BigIron RX(config)# ipv6 prefix-list filterOspfRoutesVe seq 5 deny 3015::/64
BigIron RX(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.

```
BigIron RX(config)# ipv6 router ospf
BigIron RX(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.

```
BigIron RX# 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.

```
BigIron RX(config)# route-map allowInternalRoutes permit 10
BigIron RX(config-routemap allowInternalRoutes)# match route-type internal
```

Refer to Chapter 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.

```
BigIron RX(config)# ipv6 router ospf
BigIron RX(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.

```
BigIron RX# 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 Brocade 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 Brocade 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 Brocade 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.

```
BigIron RX(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 1208

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 you specify 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 Brocade device uses the following timers when calculating the shortest path for OSPFv3 routes:

- **SPF delay** – When the Brocade 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 Brocade device waits a specific amount of time between consecutive SPF calculations. By default, the device 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 Brocade device does not accept only one timer value.

To change the SPF delay to 10 seconds and the SPF hold to 20 seconds, enter the following command.

```
BigIron RX(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 Brocade 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 Brocade 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.

```
BigIron RX(config-ospf6-router)# distance intra-area 80
BigIron RX(config-ospf6-router)# distance inter-area 90
BigIron RX(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 Brocade 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 Brocade device refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the Brocade 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 Brocade 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 Brocade 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.

```
BigIron RX(config)# ipv6 router ospf
BigIron RX(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 Brocade 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.

```
BigIron RX(config-ospf6-router)# database-overflow-interval 60
```

Syntax: [no] auto-cost reference-bandwidth <*number*>

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.

```
BigIron RX(config-ospf6-router)# external-lsdb-limit 3000
```

Syntax: ipv6 ospf area <*number*> | <*ipv4-address*>

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 value can be from 0 – 3600 seconds. The default is 1 second.

Disabling or reenabling event logging

OSPFv3 does not currently support the generation of SNMP traps. Instead, you can disable or re-enable the logging of OSPFv3-related events such as neighbor state changes and database overflow conditions. By default, the Brocade device logs these events.

To disable the logging of events, enter the following command.

```
BigIron RX(config-ospf6-router)# no log-status-change
```

Syntax: [no] log-status-change

To re-enable the logging of events, enter the following command.

```
BigIron RX(config-ospf6-router)# log-status-change
```

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

Displaying OSPFv3 area information

To display global OSPFv3 area information for the Brocade device, enter the following command at any CLI level.

```
BigIron RX# 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 204 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	Number of LSAs with a scope of the specified area.
SPF algorithm executed	The number of times 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.
Area	The area number.
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 Brocade 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.

```
BigIron RX# show ipv6 ospf database
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Link 000001e6 223.223.223.223 800000ab 1547 8955   68
0               Link 000000d8 1.1.1.1         800000aa 1295 0639   68
0               Link 00000185 223.223.223.223 800000ab 1481 7e6b   56
0               Iap  00000077 223.223.223.223 800000aa 1404 966a   56
0               Rtr  00000124 223.223.223.223 800000b0 1397 912c   40
0               Net  00000016 223.223.223.223 800000aa 1388 1b09   32
0               Iap  000001d1 223.223.223.223 800000bd 1379 a072   72
0               Iap  000000c3 1.1.1.1         800000ae 1325 e021   52
0               Rtr  00000170 1.1.1.1         800000ad 1280 af8e   40
N/A             Extn 00000062 223.223.223.223 800000ae 1409 0ca7   32
N/A             Extn 0000021d 223.223.223.223 800000a8 1319 441e   32
```

Syntax: show ipv6 ospf database [advrtr <*ipv4-address*> | as-external | extensive | inter-prefix | inter-router | intra-prefix | link | link-id <*number*> | network | router [scope <*area-id*> | as | link]]

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 205 OSPFv3 database summary fields

This field...	Displays...
Area ID	The OSPF area in which the Brocade 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 the LSA, in hexadecimal, from which the device learned this route.
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 Brocade 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 Brocade 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.

```
BigIron RX# 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 206 OSPFv3 detailed database information fields

This field...	Displays...
Router LSA (Type 1) (Rtr) fields	
Capability Bits	A bit that indicates the capability of the Brocade 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 Brocade router ID is the IPv4 address configured on the lowest numbered loopback interface. If the Brocade device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.)

TABLE 206 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 206 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 OSPFv3 interface information

You can display a summary of information for all OSPFv3 interfaces or detailed information about a specified OSPFv3 interface.

To display a summary of OSPFv3 interfaces, enter the following command at any CLI level.

```
BigIron RX# show ipv6 ospf interface
Interface  OSPF      Status State     Area
ethe 3/1             up
ethe 3/2   enabled   up     DR        0
ethe 3/4   disabled  down
loopback 2 enabled   up     Loopback  0
tunnel 1   disabled  down
tunnel 2   enabled   up     P2P       0
tunnel 6             up
```

Syntax: show ipv6 ospf interface [ethernet <*port*> | loopback <*number*> | tunnel <*number*> | ve <*number*>]

The **ethernet | 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 207 Summary of OSPFv3 interface information

This field...	Displays...
Interface	The interface type, and the port number or number of the interface.
OSPF	The state of OSPFv3 on the interface. Possible states include the following: • Enabled. • Disabled.
Status	The status of the link. Possible status include the following: • Up. • Down.
State	The state of the interface. Possible states includes 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.
Area	The OSPF area to which the interface belongs.

For example, to display detailed information about Ethernet interface 2, enter the following command at any level of the CLI.

```
BigIron RX# show ipv6 ospf interface ethernet 3/2
ethe 3/2 is up, type BROADCAST
  IPv6 Address:
      2002:c0a8:46a::1/64
      2000:4::106/64
  Instance ID 0, Router ID 223.223.223.223
  Area ID 0, Cost 1
  State DR, Transmit Delay 1 sec, Priority 1
  Timer intervals :
    Hello 10, Dead 40, Retransmit 5
  DR:223.223.223.223 BDR:1.1.1.1  Number of I/F scoped LSAs is 2
  DRElection:     5 times, DelayedLSAck:   523 times
  Neighbor Count = 1,   Adjacent Neighbor Count= 1
     Neighbor:
      1.1.1.1 (BDR)
    Statistics of interface ethe 3/2:
      Type       tx   rx tx-byte rx-byte
      Unknown     0    0       0       0
      Hello    3149 3138 1259284 1255352
      DbDesc      7    6     416     288
      LSReq       2    2      80     152
      LSUpdate 1508  530  109508   39036
      LSAck     526 1398   19036   54568
```

This display shows the following information.

TABLE 208 Detailed OSPFv3 interface information

This field...	Displays...
Interface status	The status of the interface. 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
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 Brocade device. By default, the Brocade 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.
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.

TABLE 208 Detailed OSPFv3 interface information (Continued)

This field...	Displays...
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.

```
BigIron RX# 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 209 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 Brocade's 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 Brocade 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.

```
BigIron RX# 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 210 Summary of OSPFv3 neighbor information

Field	Description
Router ID	The IPv4 address of the neighbor. By default, the Brocade 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 Brocade 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.

TABLE 210 Summary of OSPFv3 neighbor information (Continued)

Field	Description
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.

```
BigIron RX# 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 211 Detailed OSPFv3 neighbor information

Field	Description
Router ID	For information about this field, refer to Table 210 on page 1228.
Pri	For information about this field, refer to Table 210 on page 1228.
State	For information about this field, refer to Table 210 on page 1228.
DR	For information about this field, refer to Table 210 on page 1228.
BDR	For information about this field, refer to Table 210 on page 1228.
Interface [State]	For information about this field, refer to Table 210 on page 1228.

TABLE 211 Detailed OSPFv3 neighbor information (Continued)

Field	Description
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.
BadLSReq	The number of times the neighbor received a bad link-state request from the Brocade 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 Brocade device.
LSUpdateRetrans	The number of times the neighbor retransmitted link-state updates to the Brocade device.
LSA Received	The number of times the neighbor received LSAs from the Brocade device.
LS Update Received	The number of times the neighbor received link-state updates from the Brocade device.

Displaying routes redistributed into OSPFv3

You can display all IPv6 routes or a specified IPv6 route that the Brocade 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.

```
BigIron RX# 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.

```
BigIron RX# 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 212 OSPFv3 redistribution information

This Field...	Displays...
ID	An ID for the redistributed route.
Prefix	The IPv6 routes redistributed into OSPFv3.
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 Brocade 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.

```
  BigIron RX# show ipv6 ospf routes
Current Route count: 4
    Intra: 4 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 2000:4::/64                    V6E---R-- 0.0.0.0            1 0
    ::                             ethe 3/2
*IA 2002:c0a8:46a::/64             V6E---R-- 0.0.0.0            1 0
    ::                             ethe 3/2
*IA 2999::1/128                    --------- 0.0.0.0            0 0
    ::                             loopback 2
*IA 2999::2/128                    V6E---R-- 0.0.0.0            1 0
    fe80::2e0:52ff:fe91:bb37       ethe 3/2
```

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.

```
 BigIron RX# show ipv6 ospf routes 2000:4::
 Destination                    Options   Area            Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 2000:4::/64                    V6E---R-- 0.0.0.0            1 0
    ::                             ethe 3/2
```

These displays show the following information.

TABLE 213 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 Brocade device can forward IPv6 packets. "*IA" indicates the next router is an intra-area router.

TABLE 213 OSPFv3 route information (Continued)

This field...	Displays...
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.
Area	The area whose link state information has led to the routing table entry's collection of paths.
Cost	The type 1 cost of this route.
Type2 Cost	The type 2 cost of 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.

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.

```
BigIron RX# 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 214 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.

```
BigIron RX# 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 215 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 Brocade 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.

```
BigIron RX# 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 for the Brocade device, enter the following command at any level of the CLI.

```
BigIron RX# 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 216 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	IPv4 address 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 Brocade device, enter the following command at any level of the CLI.

```
BigIron RX# show ipv6 ospf virtual-neighbor
Index Router ID        Address                    State     Interface
1     1.1.1.1          3002::1                    Full      ethe 2/3
```

Syntax: show ipv6 ospf virtual-neighbor

This display shows the following information.

TABLE 217 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 217 OSPFv3 virtual neighbor information (Continued)

This field...	Displays...
State	The state between the Brocade 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.

Chapter

Configuring IPv6 Multicast Features

49

In this chapter

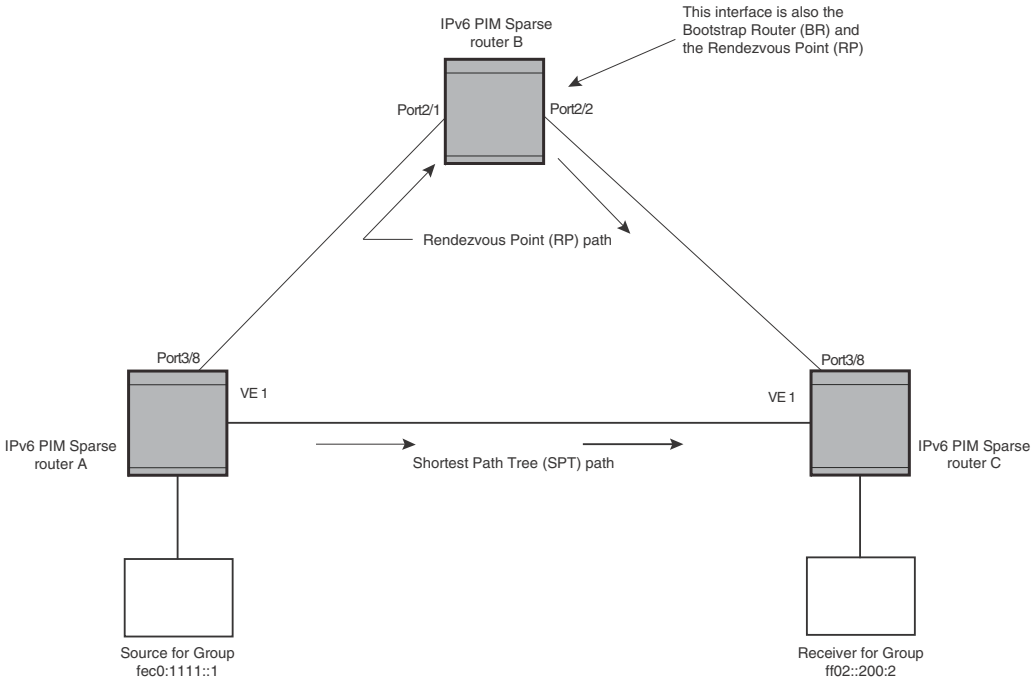
IPv6 PIM sparse

This chapter presents the multicast features available for IPv6 routers.

The BigIron RX supports IPv6 Protocol Independent Multicast (PIM) Sparse. 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 139 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 139, PIM Sparse router B is the BSR. Port 2/2 is configured as a candidate BSR.
- **RP** – The RP is the rendezvous 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 139, PIM Sparse router B is the RP. Port 2/2 is configured as a candidate Rendezvous Point (RP).

  To enhance overall network performance, Brocade BigIron RX use the RP to forward only the first packet from a group source to the group's receivers. After the first packet, the BigIron RX 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 BigIron RX calculates a separate SPT for each source-receiver pair.

NOTE
Brocade recommends that you configure the same ports as candidate BSRs and RPs.

RP paths and SPT paths

Figure 139 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 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 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 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 BigIron RX forward 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 139, the BigIron RX A forwards the first packet from group fec0:1111::1's 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.

Configuring PIM sparse

To configure a BigIron RX for IPv6 PIM Sparse, perform the following tasks:

- Configure the following global parameter

- Enable the IPv6 PIM Sparse mode of multicast routing
- Enable the IPv6 unicast-routing

- Configure the following interface parameters:
  - Configure an IPv6 address on the interface
  - Enable IPv6 PIM Sparse
  - Identify the interface as a IPv6 PIM Sparse border, if applicable

NOTE
You cannot configure a Brocade routing interface as a PMBR interface for PIM Sparse in the current software release.

- Configure the following PIM Sparse global parameters:
  - Identify the BigIron RX as a candidate PIM Sparse Bootstrap Router (BSR), if applicable.
  - Identify the BigIron RX 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
Brocade recommends that you configure the same BigIron RX as both the BSR and the RP.

IPv6 PIM-sparse mode

To configure a Brocade 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.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)#
```

Syntax: [no] ipv6 router pim

To enable IPv6 PIM Sparse mode on an interface, enter a commands such as the following.

```
BigIron RX(config)# interface ethernet 2/2
BigIron RX(config-if-e10000-2/2)# ipv6 address a000:1111::1/64
BigIron RX(config-if-e10000-2/2)# ipv6 pim-sparse
```

Syntax: [no] ipv6 pim-sparse

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 a command such as the following.

```
BigIron RX(config)#interface ve 15
BigIron RX(config-vif-15)ipv6 address a000:1111::1/64
BigIron RX(config-vif-15)#ipv6 pim-sparse
```

Configuring BSRs

In addition to the global and interface parameters in the sections above, you need to identify an interface on at least one BigIron RX as a candidate PIM Sparse Bootstrap router (BSR) and candidate PIM Sparse Rendezvous Point (RP).

NOTE
It is possible to configure the BigIron RX as only a candidate BSR or RP, but Brocade recommends that you configure the same interface on the same BigIron RX as both a BSR and an RP.

To configure the BigIron RX as a candidate BSR, enter commands such as the following.

```
BigIron RX(config)#ipv6 router pim
BigIron RX(config-ip6-pim-router)#bsr-candidate eth 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.

Syntax: [no] bsr-candidate ethernet <*slot*>/<*portnum*> | loopback <*num*> | ve <*num*> <*hash-mask-length*> [<*priority*>]

The **ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> parameter specifies the interface. The BigIron RX will advertise the specified interface's IP address 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.

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.

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.

Setting the BSR message interval

This parameter defines the interval at which the BSR sends RP candidate data to all IPv6 enabled routers within the IPv6 PIM Sparse domain. Default is 60 seconds.

To set the IPv6 PIM BSR message interval timer to 16 seconds, enter commands such as the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# 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.

Configuring RPs

Enter a command such as the following to configure the BigIron RX as a candidate RP.

```
BigIron RX(config)#ipv6 router pim
BigIron RX(config-ipv6-pim-router)# rp-candidate ethernet 2/2
```

Syntax: [no] rp-candidate ethernet <*slot*>/<*portnum*> | loopback <*num*> | ve <*num*> l pos <*slot*>/<*portnum*>

The **ethernet** <*slot*>/<*portnum*> | **loopback** <*num*> | **ve** <*num*> parameter specifies the interface. The BigIron RX will advertise the specified interface's IP address 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.

To add address ranges for which the BigIron RX is a candidate RP, enter commands such as the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# rp-candidate add ff02::200:2 64
```

Syntax: [no] rp-candidate add <*group-ipv6 addr*> <*mask-bits*>

You can change the group numbers for which the BigIron RX is a candidate RP by deleting address ranges. To delete a RP candidate, enter commands such as the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# rp-candidate delete ff02::200:1 128
```

Syntax: [no] rp-candidate delete <*group-ipv6 addr*> <*mask-bits*>

The usage of the <*group-ipv6 addr*> <*mask-bits*> parameter is the same as for the **rp-candidate add** command.

Statically specifying the RP

Brocade recommends that you use the IPv6 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 its IPv6 address, use the **rp-address** command.

If you explicitly specify the RP, the BigIron RX 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 routers within the PIM Sparse domain. Make sure the router 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.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)#rp-address 31::207
```

The command in the example above identifies the router interface at IPv6 address 31::207 as the RP for the IPv6 PIM Sparse domain. The BigIron RX will use the specified RP and ignore group-to-RP mappings received from the BSR.

Syntax: [no] rp-address <*ipv6-addr*>

The <*ipv6-addr*> parameter specifies the IPv6 address of the RP.

ACL based RP assignment

The BigIron RX, 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 ff00::/8. 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 will be 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 32 static RPs in the systems.

Configuring an ACL based RP assignment

To configure an ACL based RP assignment; enter commands such as the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# rp-address 2000::16 acl1
```

Syntax: rp-address <*ipv6_address*> [<*acl_name_or_id*>]

Use the **ipv6 address** parameter to specify the IPv6 address of the router you want to designate as an RP router.

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.

Displaying the static RP

Use the **show ipv6 pim rp-set** command to display static RP and the associated group ranges.

```
BigIron RX(config)# show ipv6 pim rp-set
Number of group-to-RP mappings: 2

      Group address         RP address
-------------------------------------------------
   1  ff03::2               2000::16
   2  ff78:440:2000::1      2000::4
```

Syntax: show ipv6 pim rp-set

Use the **show ipv6 pim rp-map** command to display all current multicast group addresses to RP address mapping.

```
BigIron RX(config-ipv6-pim-router)#sho ipv6 pim rp-map
Static RP and associated group ranges
-------------------------------------

Static RP count: 1

2000::16

Number of group prefixes Learnt from BSR: 1

Group prefix = ff00::/8      # RPs: 3
      RP 1: 2000::8    priority=0    age=30
      RP 2: 2000::4    priority=0    age=50
      RP 3: 2000::16    priority=0    age=20
```

Syntax: show ipv6 pim rp-set

Updating IPv6 PIM-sparse forwarding entries with 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 new RP configuration, enter the following command at the privileged EXEC level of the CLI.

```
BigIron RX(config)# clear ipv6 pim rp-map
```

Syntax: clear ipv6 pim rp-map

Embedded Rendezvous Point (RP)

Global deployment of IPv4 Multicast within multiple PIM Sparse domain relies on MSDP to convey information about the active sources. Since 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 command:

```
BigIron RX(config)#no rp-embedded
```

Syntax: [no] rp-embedded

Changing the Shortest Path Tree (SPT) threshold

In a typical IPv6 PIM Sparse domain, there may be two or more paths from a DR (designated router) for a multicast source to an IPv6 PIM group receiver.

- **Path through the RP** – This is the path the BigIron RX 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 BigIron RX to the receiver.
- **Shortest Path** – Each IPv6 PIM Sparse router ithat is a DR for an IPv6 receiver calculates a short path tree (SPT) towards the source of the IPv6 multicast traffic. The first time a BigIron RX that is configured as an IPv6 PIM router receives a packet for an IPv6 group, it sends the packet to the RP for hat group, which in turn will forward it to all the intended DRs that have registered with the RP. The first time a BigIron RX that is a recipient receives a packet for an IPv6 group, it evaluates the shortest path to the source and initiates a switch over to the SPT. Once the BigIron RX starts receiving data on the SPT, the BigIron RX will proceed 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 BigIron RX maintains a separate counter for each IPv6 PIM Sparse source-group pair.

You can change the number of packets that the BigIron RX receives using the RP before switching to using the SPT.

To change the number of packets the BigIron RX receives using the RP before switching to the SPT, enter commands such as the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-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 BigIron RX sends packets using the RP indefinitely and does not switch over to the SPT. If you enter a specific number of packets, the BigIron RX 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 BigIron RX sends candidate RP advertisement messages to the BSR, enter commands such as the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# 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.

Changing the PIM join and prune message interval

By default, the BigIron RX sends PIM Sparse Join/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/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/Prune interval, enter commands such as the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# message-interval 30
```

Syntax: [no] message-interval <*seconds*>

The <*seconds*> parameter specifies the number of seconds and can be from 1 – 65535. The default is 60 seconds.

Setting the inactivity timer

The router deletes a forwarding entry if the entry is not used to send multicast packets. TheIPv6 PIM inactivity timer defines how long a forwarding entry can remain unused before the router deletes it.

To apply a IPv6 PIM inactivity timer of 160 seconds to all IPv6 PIM interfaces, enter the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# inactivity-timer 160
```

Syntax: [no] inactivity-timer <*seconds*>

The <*seconds*> parameter specifies the number of seconds. Valid range is 60 - 3600. The default is 180 seconds.

Changing the hello timer

This parameter defines the interval at which periodic hellos are sent out 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.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)#hello-timer 62
```

Syntax: [no] hello-timer <*seconds*>

The <*seconds*> parameter specifies the number of seconds. Valid range is 10 - 3600. 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 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 ff30::/12 has been assigned by the Internet Assigned Numbers Authority (IANA) for use with SSM.

SSM simplifies PIM-SM by eliminating the RP and all protocols related to the RP.

Configuring source-specific multicast

PIM-SM must be enabled on any ports that you want SSM to operate. Enter the **ssm-enable** command under the IPv6 router PIM level to globally enable source specific multicast filtering.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# ssm-enable
```

Syntax: [no] ssm-enable

Displaying IPv6 PIM-sparse configuration information

To display IPv6 PIM Sparse configuration information, use the **show ipv6 pim sparse** command as described in “Displaying IPv6 PIM-sparse configuration information” on page 1248.

Passive Multicast Route Insertion (PMRI)

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 Switches running software release 03.6.00 and later.

PMRI enables a Layer 3 switch running IPv6 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 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 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

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.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# 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.

```
BigIron RX#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
```

Syntax: show ipv6 pim sparse

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 IPv6 PIM Sparse group
- RP set list
- IPv6 PIM Neighbor information
- The IPv6 PIM flow cache
- The IPv6PIM multicast cache
- IPv6 PIM traffic statistics
- IPv6 PIM counter statistics

Displaying basic PIM sparse configuration information

To display IPv6 PIM Sparse configuration information, enter the following command at any CLI level.

```
BigIron RX#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 sparse

This display shows the following information.

TABLE 218

This field...	Displays...
Global PIM sparse mode settings	
Hello interval	How frequently the BigIron RX sends IPv6 PIM Sparse hello messages to its IPv6 PIM Sparse neighbors. This field show the number of seconds between hello messages. IPv6 PIM Sparse routers use hello messages to discover one another.
Neighbor timeout	How many seconds the BigIron RX 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 BigIron RX 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. A candidate RP's group prefix 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 BigIron RX is elected to be the BSR. Otherwise, the field is blank.
Candidate-RP Advertisement interval	How frequently the candidate PR configured on the BigIron RX sends candidate RP advertisement messages to the BSR. **NOTE:** This field contains a value only if an interface on the BigIron RX is configured as a candidate RP. Otherwise, the field is blank.
Join/Prune interval	How frequently the BigIron RX sends IPv6 PIM Sparse Join/Prune messages for the multicast groups it is forwarding. This field show the number of seconds between Join/Prune messages. The BigIron RX sends Join/Prune messages on behalf of multicast receivers who want to join or leave an IPv6 PIM Sparse group. When forwarding packets from IPv6 PIM Sparse sources, the BigIron RX sends the packets only on the interfaces on which it has received join requests in Join/Prune messages for the source's group.
SPT Threshold	The number of packets the BigIron RX 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 router.
IPv6 PIM sparse interface information	

NOTE: You also can display IPv6 multicast interface information using the show ipv6 pim interface command.

TABLE 218

This field...	Displays...
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
Local Address	Indicates the IP address configured on the port or virtual interface.

Displaying IPv6 PIM inteface information

You also can display IPv6 multicast interface information using the **show ipv6 pim interface** command.

```
BigIron RX# 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
```

Displaying a list of multicast groups

To display IPv6 PIM group information, enter the following command at any CLI level.

```
BigIron RX>show ipv6 pim group
Total number of groups: 11
Group ff7e:a40:2001:3e8:27:0:1:2  Ports
     Group member at  e3/1: v31
```

Syntax: show ipv6 pim [group

This display shows the following information.

This field...	Displays...
Total number of Groups	Lists the total number of IPv6 multicast groups the BigIron RX is forwarding. **NOTE:** This list can include groups that are not IPv6 PIM Sparse groups. If interfaces on the BigIron RX are configured for regular Ipv6 PIM (dense mode) or DVMRP, these groups are listed too.
Group	The multicast group address
Ports	The BigIron RX ports connected to the receivers of the groups.

Displaying BSR information

```
BigIron RX#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
BigIron RX#
```

This example show information displayed on a BigIron RX that has been elected as the BSR. The following example shows information displayed on a BigIron RX that is not the BSR. Notice that some fields shown in the example above do not appear in the example below.

```
BigIron RX>show ipv6 pim bsr
PIMv2 Bootstrap information
 BSR address  = 2001:3e8:255:255::17
 BSR priority = 0
BigIron RX>
```

Syntax: show ipv6 pim bsr.

This display shows the following information.

This field...	Displays...
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 BigIron RX 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.

This field...	Displays...
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 BigIron RX can be a BSR. The default is 32 bits, which allows the BigIron RX to be a BSR for any valid IPv6 multicast group number. **NOTE:** This field appears only if this BigIron RX 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 BigIron RX is the BSR.
Next Candidate-RP-advertisement message in	Indicates how many seconds will pass before the BSR sends its next candidate PR advertisement message. **NOTE:** This field appears only if this BigIron RX is a candidate BSR.
RP	Indicates the IPv6 address of the Rendezvous Point (RP). **NOTE:** This field appears only if this BigIron RX 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 BigIron RX 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 BigIron RX is a candidate BSR.

Displaying candidate RP information

To display candidate RP information, enter the following command at any CLI level.

```
BigIron RX# 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 show information displayed on a BigIron RX that is a candidate RP. The following example shows the message displayed on a BigIron RX that is not a candidate RP.

```
BigIron RX# show ipv6 pim rp-candidate
```

This system is not a Candidate-RP.

Syntax: show ipv6 pim rp-candidate

This display shows the following information.

This field...	Displays...
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 BigIron RX is a candidate RP.
RP	Indicates the IPv6 address of the Rendezvous Point (RP). **NOTE:** This field appears only if this BigIron RX is a candidate RP.

This field...	Displays...
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 BigIron RX 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 BigIron RX is a candidate RP.

Displaying RP-to-group mappings

To display RP-to-group-mappings, enter the following command at any CLI level.

```
BigIron RX# show ipv6 pim rp-map
Idx Group address                              RP address
------------------------------------------------------------------------------
  1                                 ff1e::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
BigIron RX#
```

Syntax: show ipv6 pim rp-map

This display shows the following information.

This field...	Displays...
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 a PIM sparse group

To display RP information for a PIM Sparse group, enter the following command at any CLI level.

```
BigIron RX#show ipv6 pim rp-hash ff1e::1:2
  RP: 2001:3e8:255:255::17, v2
    Info source: 2001:3e8:255:255::17, via bootstrap
BigIron RX#
```

Syntax: show ipv6 pim rp-hash <*group-addr*>

The <*group-addr*> parameter is the address of an IPv6 PIM Sparse IP multicast group.

This display shows the following information.

This field...	Displays...
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 BigIron RX 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 BigIron RX 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.

```
BigIron RX#show ipv6 pim rp-set

Number of group prefixes Learnt from BSR: 1

Group prefix = ff00::/8      # RPs expected: 1
      # RPs received: 1
      RP 1: 2001:3e8:255:255::17    priority=0    age=0
BigIron RX#
```

Syntax: show ipv6 pim rp-set

This display shows the following information.

This field...	Displays...
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/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 BigIron RX is not a BSR, this field contains zero. Only the BSR ages the RP-set.

Displaying multicast neighbor information

To display information about the BigIron RX's IPv6 PIM neighbors, enter the following command at any CLI level.

```
BigIron RX#show ipv6 pim nbr
Port Phy_Port Neighbor                                  Holdtime Age   UpTime
                                                        sec      sec   sec
e11/15 e11/15 fe80::45:27:49:4                          105        20    1010
v312   e11/3  fe80::45:27:1:2                           105        10    1900

BigIron RX#
```

Syntax: show ipv6 pim nbr

This display shows the following information.

This field...	Displays...
Port	The interface through which the BigIron RX is connected to the neighbor.
Neighbor	The IPv6 interface of the IPv6 PIM neighbor interface.
Phy_Port	

This field...	Displays...
Holdtime sec	Indicates how many seconds the neighbor wants this BigIron RX to hold the entry for this neighbor in memory. The neighbor sends the Hold Time in its Hello packets. • If the BigIron RX receives a new Hello packet before the Hold Time received in the previous packet expires, the BigIron RX updates its table entry for the neighbor. • If the BigIron RX does not receive a new Hello packet from the neighbor before the Hold time expires, the BigIron RX assumes the neighbor is no longer available and removes the entry for the neighbor.
Age sec	The number of seconds since the BigIron RX 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 BigIron RX receives the first Hello messages from the neighbor.

Displaying the IPv6 PIM multicast cache

To display the IPv6 PIM multicast cache, enter the following command at any CLI level.

```
BigIron RX# 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 mcache

```
Displaying the IPv6 PIM Resources
```

To display the hardware resource information such as hardware allocation, availability, and limit for software data structure, enter the following command.

```
BigIron RX>show ipv6 pim traffic
Port    Hello           J/P          Register       RegStop          Assert
     [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]
MLD Statistics:
  Total Recv/Xmit 0/0
  Total Discard/chksum  0/0
```

This display shows the following information.

TABLE 219 Output of Show IPvG PIM resource

This field...	Displays...
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 following command at any CLI level.

```
BigIron RX#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
BigIron RX#
```

Syntax: show ipv6 pim traffic

This display shows the following information.

This field...	Displays...
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/P	The number of Join/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/Xmit	The total number of IGMP messages sent and received by the BigIron RX.
Total Discard/chksum	The total number of IGMP messages discarded, including a separate counter for those that failed the checksum comparison.

Multicast Listener Discovery and source specific multicast protocols(MLDv2)

The Multicast Listener Discovery Version 2 (MLDv2) protocol is available on the BigIron RX that is running IPv6. 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 switch stores a list of multicast addresses for each attached link. For each multicast address, the IPv6 switch 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 o 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.

NOTE
PIM IPv4/IPv6 can only be configured on one physical port. The loopback interface does not count as one physical interface. The chassis will give you an error if you attempt to configure PIM on more than on physical interface.

NOTE
IPv6 will not be work unless all line cards are on the same version. If one LP is not the IPv6 version, IPv6 will not work on the chassis.

MLD version distinctions

MLDv1 mode interfaces exchange the following types of messages between routers and hosts:

- Multicast listener queries

  - General Query, used to learn which multicast addresses have listeners on an attached link.
  - Multicast-Address-Specific Query, used to learn if a particular multicast address has any listeners on an attached link.
- Multicast listener reports
- Multicast listener done messages

MLDv2 mode interfaces exchange the following types of messages with MLDv2 hosts:

  - Multicast listener queries
  - MLDv2 multicast listener reports

Enabling MLDv2

MLDv1 is enabled once PIM Sparse (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 PIM-SM, do the following.

1. Enable IPv6 PIM mode by entering a command such as the following.

   ```
   BigIron RX(config)#ipv6 router pim
   BigIron RX(config-ipv6-pim-router)#
   ```

   Syntax: [no] ipv6 router pim

2. At the interface level enable MLDv2 by entering the following commands.

   ```
   BigIron RX(config)#interface ethernet 1/1
   BigIron RX(config-if-e10000-1/1)#ipv6 pim-sparse
   ```

   Syntax: [no] ipv6 pim-sparse

3. Once PIM-SM is enabled, specify which version of MLD will be used by entering the following command.

   ```
   BigIron RX(config-if-e10000-1/1)#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.

Enabling source specific multicast

Once MDLv2 is enabled, source specific multicast for PIM can be enabled for multicast group addresses in the ff30::0/16 IPv6 address range. If MLDv2 is enabled, but SSM is not, the IPv6 router builds the Shortest Path Tree (SPT) as well as the shared (RP) tree and produces a (*.G) record. If both MLDv2 and SSM are enabled, then only the SPT is built to produce a (S,G) record.

To enable SSM on a Brocade device running PIM-SM, enter commands such as the following.

```
BigIron RX(config)# ipv6 router pim
BigIron RX(config-ipv6-pim-router)# ssm-enable
```

Syntax: [no] ssm-enable

Enter the **ssm-enable** command under the IPv6 router PIM level to globally enable source specific multicast filtering.

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.

```
BigIron RX(config)#ipv6 mld query-interval 50
```

Syntax: ipv6 mld query-interval <*seconds*>

Specify 1 – 3600 for <*seconds*>. The default is 60 seconds.

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.

```
BigIron RX(config)#ipv6 mld max-response-time 5
```

Syntax: ipv6 mld max-response-time <*seconds*>

Specify 1 – 64 for <*seconds*>. The default is 5 seconds.

Setting the last listener query count

The Last Listener Query Count is the number of Multicast-Address- Specific Queries sent before the switch assumes there are no remaining listeners for an address on a link. You can set the last listener query count by entering a command such as the following.

```
BigIron RX(config)#ipv6 mld llqc 5
```

Syntax: ipv6 mld llqc <*seconds*>

Specify 2 – 7 for <*seconds*>.

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 listenener query interval by entering a command such as the following.

```
BigIron RX(config)#ipv6 mld llqi 5
```

Syntax: ipv6 mld llqi <*seconds*>

Specify 1 – 10 for <*seconds*>.

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.

```
BigIron RX(config)#ipv6 mld robustness 3
```

Syntax: ipv6 mld robustness<*seconds*>

Specify 2 – 7 for <*seconds*>. Default is 2

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.

```
BigIron RX(config)#ipv6 mld version 2
```

Syntax: ipv6 mld version <*version-number*>

Enter 1or 2 for <*version-number*> Default version 2

Specifying a port version

At the interface level, you can specify the MLD version for a physical port within a virtual interface. You can set the version by entering a command such as the following at the interface level.

```
BigIron RX(config-vif-401)#ipv6 mld port-ver 1 eth 3/1
```

Syntax: mld port-ver <*version-number*>

Specify 1 or 2 for <*version-number*>.

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.

```
BigIron RX(config-vif-401))#ipv6 mld static-group ffe0::4c8 eth 3/1
To configure a Static Group on a physical interface, enter a command such as the
following:
BigIron RX (config-if-e1000-5/23)#ipv6 mld static ff01::6f
```

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 number of the port that will be included in this static group for the ethernet <*port-number*> parameter. The asterisk (*) in the syntax above 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.

Setting the interface MLD version

You can use this command to set the MLD version (1 or 2) for the interface. You can select the version of MLD by entering a command such as the following at the interface level.

```
BigIron RX(config-lbif-1)#ipv6 mld version
```

Syntax: ipv6 mld version <*version-number*>

Enter 1or 2 for <*version-number*>. Default is version 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.

```
BigIron RX #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
```

This field...	Displays...
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.
ipv6 address	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

Syntax: show ipv6 mld group

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.

```
BigIron RX #show ipv6 mld interface
version = 2, query int = 125, max resp time = 10, group mem time = 635
robustness = 5, other querier present time = 630
last listener query int = 1, last listener query count = 5
e5/18: default V2, PIM sparse, addr=fe80::20c:dbff:fe80:5251
has 50 groups, Querier, default V2
group: ff1e::619, exclude, permit 0, exclude-time=513, deny 0
group: ff1e::618, exclude, permit 0, exclude-time=513, deny 0
group: ff1e::617, exclude, permit 0, exclude-time=513, deny 0
group: ff1e::616, exclude, permit 0, exclude-time=513, deny 0
group: ff1e::5a8, exclude, permit 0, exclude-time=513, deny 0
group: ff1e::5a7, exclude, permit 0, exclude-time=513, deny 0
group: ff1e::5a6, exclude, permit 0, exclude-time=513, deny 0
```

This field...	Displays...
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.

Syntax: show ipv6 mld interface [<*port-number*>]

Enter the port's number if you want to display MLD information for a specific interface.

Displaying MLD traffic

To display information on MLD traffic, enter a command such as the following.

```
BigIron RX #show ipv6 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. These sections show the following information.

This field	Displays
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.

Syntax: show ipv6 mld traffic

Clearing IPv6 MLD traffic

To clear statistics on IPv6 MLD traffic, enter the following command.

```
BigIron RX(config)#clear ipv6 mld traffic ethernet 7/10
```

Syntax: clear ipv6 mld traffic ethernet <*slot-number*>/<*port-number*> | ve <*ve-number*>

Select **ethernet** and enter the interface's slot number and port number to clear MLD traffic on a physical interface.

Embedded Rendezvous Point (RP)

Release 02.6.00 provides support for embedded RP which enables a router to learn the RP information using the multicast group destination address. This eliminates the need to maintain the RP table through the RP state machine. For routers that are the RP, the router is statically configured as the RP. This router is not required to be configured as a RP candidate.

A designated router (DR) receiving a MLD report checks for the presence of an embedded RP address in the group address. If found, it uses this RP address as the target of the PIM Join that it initiates and all subsequent protocol operations for this group.

A designated router (DR) receiving a new multicast stream from a directly connected source, checks for the presence of an embedded RP address in the group address. If found, it uses this RP address as the unicast destination address of the PIM Register packets that it initiates.

Other PIM routers that are also receiving PIM PDUs look into the group address for the presence of an embedded RP address in the group address. If found, it uses this RP address for all its protocol needs.

Note that all 3 cases above assume that a static RP has not been configured locally. A statically configured RP has the highest precedence and will be used even if the group address contains an embedded RP address. When determining the RP, the software employs the following precedence table with the highest precedence item listed first.

1. Statically configured RP address
2. Embedded RP address
3. RP Set build using the RP state machine

Enabling the embedded RP

The following command may be used to enable the embedded RP feature.

```
BigIron RX(config-ipv6-pim-router)# rp-embedded
```

Syntax: [no] rp-embedded

Default: On

Chapter

50

Configuring IPv6 Routes

In this chapter

Configuring a static IPv6 route

This chapter provides information on how to configure a static IPv6 route. A ***static IPv6 route*** is a manually configured route, which creates a path between two IPv6 routers. 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.

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 "Configuring a static IPv6 route" on page 1267.

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.

```
BigIron RX(config)# ipv6 route 8eff::0/32 4fee:2343:0:ee44::1 distance 110
```

Syntax: 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.

```
BigIron RX(config)# ipv6 route 8eff::0/32 ethernet 1 fe80::1
```

Syntax: ipv6 route <*dest-ipv6-prefix*>/<*prefix-length*> [ethernet <*slot/port*> | pos <*slot/port*> | ve <*num*> | null0] <*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 that the router can access through tunnel 1, enter the following command.

```
BigIron RX(config)# ipv6 route 8eff::0/32 tunnel 1
```

Syntax: ipv6 route <*dest-ipv6-prefix*>/<*prefix-length*> <*interface*> <*port*> [<*metric*>] [distance <*number*>]

Table 220 describes the parameters associated with this command and indicates the status of each parameter.

TABLE 220 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.)
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 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 BigIron RX 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.

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.

```
BigIron RX(config)# ipv6 mroute 8eff::0/32 4fee:2343:0:ee44::1
```

Syntax: ipv6 mroute <*dest-ipv6-prefix*>/<*prefix-length*> <*next-hop-ipv6-address*> <*next-hop-enable-default*> <*next-hop-recursion*> [<*metric*>] [distance <*number*>] [tag<*number*>]

Syntax: 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 BigIron RX prefers the path with the lower administrative distance.

NOTE
Regardless of the administrative distances, the switch always prefers directly connected routes over other routes.

The ipv6 mroute command is used to direct multicast traffic along a specific path. The ipv6 mroute command starts with the ipv6 address or ingress ipv6 address the source traffic is received upon. The ingress interface network mask, and the next hop address leading back to the ingress source ipv6 address.

To configure static IPv6 multicast routes, enter a command such as the following.

```
BigIron RX(config)# ipv6 mroute 12.7.1.0 255.255.255.0 17.3.1.2
```

Syntax: [no] ipv6 mroute <*ip-addr*> <*ip-mask*> [<*next-hop-ip-addr*> | ethernet <*slot/port*> | ve <*num*> | null0] [<*cost*>] [distance < *num*>]

The **ip-addr** and **ip-mask** parameters specifies the PIM source for the route.

The **ethernet** <*slot/port*> parameter specifies a physical port.

The **ve** <*num*> parameter specifies a virtual interface.

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.

Regardless of the administrative distances, the BigIron RX Series router always prefers directly connected routes over other routes.

Appendix

Using Syslog

A

This appendix describes how to display Syslog messages and how to configure the Syslog facility, and lists the Syslog messages that a BigIron RX can display during standard operation.

NOTE
This appendix does not list Syslog messages that can be displayed when a debug option is enabled.

A BigIron RX'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 1000 entries.

You also can specify the IP address or host name of up to six Syslog servers. When you specify a Syslog server, the BigIron RX 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 BigIron RX'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 BigIron RX. 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.

```
BigIron RX> 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 1273.

Enabling real-time display of Syslog messages

By default, to view Syslog messages generated by a BigIron RX, you need to display the Syslog buffer or the log on a Syslog server used by the BigIron RX.

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.

```
BigIron RX(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@BigIron RX# terminal monitor
Syslog trace was turned ON
```

Syntax: 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@BigIron RX# terminal monitor
Syslog trace was turned OFF
```

Here is an example of how the Syslog messages are displayed.

```
telnet@BigIron RX# terminal monitor
Syslog trace was turned ON
SYSLOG: <9>BigIron RX, Power supply 2, power supply on left connector, failed

SYSLOG: <14>BigIron RX, Interface ethernet 1/6, state down

SYSLOG: <14>BigIron RX, 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 BigIron RX 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 BigIron RX, enter the following command from any level of the CLI.

```
BigIron RX> 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 221 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 1278. 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 1281.
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 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.

```
BigIron RX(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.

```
BigIron RX# 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.

```
BigIron RX(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.

```
BigIron RX(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)
```

Disabling or re-enabling Syslog

Syslog is enabled by default. To disable it, enter the following command at the global CONFIG level.

```
BigIron RX(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.

```
BigIron RX(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.

```
BigIron RX(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: 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.

```
BigIron RX(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: logging host <*ip-addr*> | <*server-name*>

Disabling logging of a message level

To change the message level, disable logging of specific message levels. You must disable the message levels on an individual basis.

For example, to disable logging of debugging and informational messages, enter the following commands.

```
BigIron RX(config)# no logging buffered debugging
BigIron RX(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.

Logging all CLI commands to Syslog

This feature introduced in version 02.4.00 of the Multi-Service IronWare software 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.

```
BigIron RX(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@ BigIron RX(config)# logging cli-command
telnet@ BigIron RX(config)# router bgp
telnet@ BigIron RX(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.

```
BigIron RX(config)# router bgp
BigIron RX(config-bgp)# nei 10.1.1.8 remote 10
```

Using the **show log** command, you would see a series of log records as shown in the following.

```
BigIron RX(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
```

Changing the number of entries the local buffer can hold

You also can use the **logging buffered** command to change the number of entries the local Syslog buffer can store. For example.

```
BigIron RX(config)# logging buffered 100
```

The default number of messages is 50. The value can be from 1 – 1000. The change takes effect immediately and does not require you to reload the software.

NOTE
If you decrease the size of the buffer, the software clears the buffer before placing the change into effect. If you increase the size of the buffer, the software does not clear existing entries.

Changing the log facility

The Syslog daemon on the Syslog server uses a facility to determine where to log the messages from the BigIron RX. The default facility for messages the BigIron RX 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 BigIron RX to use two Syslog servers, the device uses the same facility on both servers.

```
BigIron RX(config)# logging facility local0
```

Syntax: 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/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/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.

```
BigIron RX(config)# ip show-portname
```

This command is applied globally to all interfaces on the BigIron RX.

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.

```
BigIron RX# 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 BigIron RX's local buffer, use the following command.

```
BigIron RX# clear logging
```

Syntax: clear logging

Displaying TCP/UDP port numbers in Syslog messages

The command **ip show-acl-service-number** allows you to change the display of TCP/UDP application information from the TCP/UDP well-known port name to the TCP/UDP port number. For example, entering the following command causes the BigIron RX 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 BigIron RX displays TCP/UDP application information in named notation.

In this release, you can display TCP/UDP port number instead of their names in syslog messages by entering the following command.

```
BigIron(config)# ip show-service-number-in-log
```

Syntax: [no]ip show-service-number-in-log

Syslog messages

Table 222 lists 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 222 Brocade **Syslog messages**

Message level	Message	Explanation
Alert	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	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.
Alert	Management module at slot <*slot-num*> state changed from <*module-state*> to <*module-state*>.	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
Alert	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.
Alert	<*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	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
Alert	ISIS MEMORY USE EXCEEDED	IS-IS is requesting more memory than is available.

TABLE 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
Alert	MAC Authentication failed for <*mac-address*> on <*portnum*> (Invalid User)	RADIUS authentication failed for the specified <*mac-address*> on the specified <*portnum*> because the MAC address sent to the RADIUS server was not found in the RADIUS server's users database.
Alert	MAC Authentication failed for <*mac-address*> on <*portnum*>	RADIUS authentication was successful for the specified <*mac-address*> on the specified <*portnum*>; however, the VLAN returned in the RADIUS Access-Accept message did not refer to a valid VLAN or VLAN ID on the BigIron RX. This is treated as an authentication failure.
Alert	MAC Authentication failed for <*mac-address*> on <*portnum*> (No VLAN Info received from RADIUS server)	RADIUS authentication was successful for the specified <*mac-address*> on the specified <*portnum*>; however, dynamic VLAN assignment was enabled for the port, but the RADIUS Access-Accept message did not include VLAN information. This is treated as an authentication failure.
Alert	MAC Authentication failed for <*mac-address*> on <*portnum*> (RADIUS given VLAN does not match with TAGGED vlan)	Multi-device port authentication failed for the <*mac-address*> on a tagged port because the packet with this MAC address as the source was tagged with a VLAN ID different from the RADIUS-supplied VLAN ID.
Alert	MAC Authentication failed for <*mac-address*> on <*portnum*> (RADIUS given vlan does not exist)	RADIUS authentication was successful for the specified <*mac-address*> on the specified <*portnum*>; however, the RADIUS Access-Accept message specified a VLAN that does not exist in the BigIron RX's configuration. This is treated as an authentication failure.
Alert	MAC Authentication failed for <*mac-address*> on <*portnum*> (Port is already in another radius given vlan)	RADIUS authentication was successful for the specified <*mac-address*> on the specified <*portnum*>; however, the RADIUS Access-Accept message specified a VLAN ID, although the port had previously been moved to a different RADIUS-assigned VLAN. This is treated as an authentication failure.
Critical	Authentication shut down <*portnum*> due to DOS attack	Denial of Service (DoS) attack protection was enabled for multi-device port authentication on the specified <*portnum*>, and the per-second rate of RADIUS authentication attempts for the port exceeded the configured limit. The BigIron RX considers this to be a DoS attack and disables the port.
Error	No of prefixes received from BGP peer <*ip-addr*> exceeds maximum prefix-limit...shutdown	The BigIron RX has received more than the specified maximum number of prefixes from the neighbor, and the BigIron RX is therefore shutting down its BGP4 session with the neighbor.

TABLE 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
Warning	Locked address violation at interface e<*portnum*>, address <*mac-address*>	Indicates that a port on which you have configured a lock-address filter received a packet that was dropped because the packet's source MAC address did not match an address learned by the port before the lock took effect. The e<*portnum*> is the port number. The <*mac-address*> is the MAC address that was denied by the address lock. Assuming that you configured the port to learn only the addresses that have valid access to the port, this message indicates a security violation.
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.
Warning	Dup IP <*ip-addr*> detected, sent from MAC <*mac-addr*> interface <*portnum*>	Indicates that the BigIron RX received a packet from another device on the network with an IP address that is also configured on the BigIron RX. 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 Brocade port that received the packet with the duplicate IP address. The address is the packet's source IP address.
Warning	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.

TABLE 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
Warning	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.
Warning	No of prefixes received from BGP peer <*ip-addr*> exceeds warning limit <*num*>	The BigIron RX 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 BigIron RX receives a 76th prefix from the neighbor.
Warning	DOT1X security violation at port <*portnum*>, malicious mac address detected: <*mac-address*>	A security violation was encountered at the specified port number.
Notification	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	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	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*>

TABLE 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
Notification	ACL exceed max DMA L4 cam resource, using flow based ACL instead	The port does not have enough Layer 4 CAM entries for the ACL. To correct this condition, allocate more Layer 4 CAM entries. To allocate more Layer 4 CAM entries, enter the following command at the CLI configuration level for the interface. **ip access-group max-l4-cam** <*num*>
Notification	ACL insufficient L4 cam resource, using flow based ACL instead	The port does not have a large enough CAM partition for the ACLs. To re-partition the CAM, refer to the "Changing CAM Partitions" chapter in the *Diagnostic Guide*.
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.
Notification	ACL 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	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 BigIron RX. 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 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
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
Notification	OSPF 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 neighbor has changed. The <router-id> is the router ID of the BigIron RX. 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 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
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 BigIron RX. 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 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 BigIron RX. The <*ip-addr*> is the IP address of the interface on the BigIron RX. The <*src-ip-addr*> is the IP address of the interface from which the BigIron RX 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 222 Brocade **Syslog messages** (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 BigIron RX. The <*ip-addr*> is the IP address of the interface on the BigIron RX. The <*src-ip-addr*> is the IP address of the interface from which the BigIron RX 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 222 Brocade **Syslog messages** (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 BigIron RX. The <*ip-addr*> is the IP address of the interface on the BigIron RX. The <*src-ip-addr*> is the IP address of the interface from which the BigIron RX 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 222 Brocade **Syslog messages** (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 BigIron RX. The <*ip-addr*> is the IP address of the interface on the BigIron RX. The <*src-ip-addr*> is the IP address of the interface from which the BigIron RX 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	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 BigIron RX. The <*ip-addr*> is the IP address of the interface on the BigIron RX. The <*src-ip-addr*> is the IP address of the interface from which the BigIron RX 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 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
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 BigIron RX. The <*ip-addr*> is the IP address of the interface on the BigIron RX. The <*src-ip-addr*> is the IP address of the interface from which the BigIron RX 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	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 BigIron RX has retransmitted a Link State Advertisement (LSA). The <*router-id*> is the router ID of the BigIron RX. The <*ip-addr*> is the IP address of the interface on the BigIron RX. 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 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
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 BigIron RX has retransmitted a Link State Advertisement (LSA). The <*router-id*> is the router ID of the BigIron RX. The <*ip-addr*> is the IP address of the interface on the BigIron RX. 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 BigIron RX. 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 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 BigIron RX. 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 BigIron RX. 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 BigIron RX. The <*num*> is the number of LSAs.

TABLE 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
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 BigIron RX's router ID. The intf addr <*ip-addr*> is the IP address of the Brocade 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.
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 BigIron RX'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.
Notification	FSRP intf state changed, intf <*portnum*>, addr <*ip-addr*>, state <*fsrp-state*>	A state change has occurred in a Foundry Standby Router Protocol (FSRP) interface. The <*portnum*> is the port. The <*ip-addr*> is the IP address of the FSRP interface. The <*fsrp-state*> can be one of the following: • init • negotiating • standby • active • unknown

TABLE 222 Brocade **Syslog messages** (Continued)

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
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 BigIron RX.
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 BigIron RX.
Notification	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 BigIron RX 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	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 BigIron RX 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	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 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
Notification	Local TCP exceeds <*num*> burst packets, stopping for <*num*> seconds!!	Threshold parameters for local TCP traffic on the device have been configured, and the maximum burst size for TCP packets has been exceeded. 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 device. **Note**: This message can occur in response to an attempted TCP SYN attack.
Notification	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.
Notification	ISIS L1 ADJACENCY DOWN <*system-id*> on circuit <*circuit-id*>	The BigIron RX'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.
Notification	ISIS L1 ADJACENCY UP <*system-id*> on circuit <*circuit-id*>	The BigIron RX'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 BigIron RX'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 BigIron RX'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 ENTERED INTO OVERLOAD STATE	The BigIron RX has set the overload bit to on (1), indicating that the BigIron RX's IS-IS resources are overloaded.
Notification	ISIS EXITING FROM OVERLOAD STATE	The BigIron RX has set the overload bit to off (0), indicating that the BigIron RX's IS-IS resources are no longer overloaded.

TABLE 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
Notification	DOT1X issues software but not physical 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 but not physical 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.
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*>.
Notification	MAC Authentication succeeded for <*mac-address*> on <*portnum*>	RADIUS authentication was successful for the specified <*mac-address*> on the specified <*portnum*>.
Informational	Cold start	The device has been powered on.
Informational	Warm start	The system software (flash code) has been reloaded.
Informational	<*user-name*> login to USER EXEC mode	A user has logged into the USER EXEC mode of the CLI. The <*user-name*> is the user name.
Informational	<*user-name*> logout from USER EXEC mode	A user has logged out of the USER EXEC mode of the CLI. The <*user-name*> is the user name.
Informational	<*user-name*> login to PRIVILEGED mode	A user has logged into the Privileged EXEC mode of the CLI. The <*user-name*> is the user name.
Informational	<*user-name*> logout from PRIVILEGED mode	A user has logged out of Privileged EXEC mode of the CLI. The <*user-name*> is the user name.
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	Interface <*portnum*>, state up	A port has come up. The <*portnum*> is the port number.
Informational	Interface <*portnum*>, state down	A port has gone down. The <*portnum*> is the port number.
Informational	Interface <*portnum*>, line protocol up	The line protocol on a port has come up. The <*portnum*> is the port number.
Informational	Interface <*portnum*>, line protocol down	The line protocol on a port has gone down. The <*portnum*> is the port number.

TABLE 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
Informational	Trunk 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 trunk group (aggregate link). The <*ports*> is a list of the ports that were aggregated to make the trunk group.
Informational	Bridge root changed, vlan <*vlan-id*>, new root ID <*string*>, root interface <*portnum*>	A Spanning Tree Protocol (STP) topology change has occurred. The <*vlan-id*> is the ID of the VLAN in which the STP topology change occurred. The <*root-id*> is the STP bridge root ID. The <*portnum*> is the number of the port connected to the new root bridge.
Informational	Bridge is new root, vlan <*vlan-id*>, root ID <*root-id*>	A Spanning Tree Protocol (STP) topology change has occurred, resulting in the BigIron RX becoming the root bridge. The <*vlan-id*> is the ID of the VLAN in which the STP topology change occurred. The <*root-id*> is the STP bridge root ID.
Informational	Bridge topology change, vlan <*vlan-id*>, interface <*portnum*>, changed state to <*stp-state*>	A Spanning Tree Protocol (STP) topology change has occurred on a port. The <*vlan-id*> is the ID of the VLAN in which the STP topology change occurred. The <*portnum*> is the port number. The <*stp-state*> is the new STP state and can be one of the following: • disabled • blocking • listening • learning • forwarding • unknown
Informational	startup configuration was changed or startup configuration was changed by <*user-name*>	A configuration change was saved to the startup configuration file. The <*user-name*> is the user's ID, if they entered a user ID to log in.
Informational	vlan <*vlan-id*> interface <*portnum*> Bridge TC Event (DOT1wTransition)	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	vlan <*vlan-id*> interface <*portnum*> STP state -> <*state*> (DOT1wTransition)	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	vlan <*vlan-id*> New RootPort <*portnum*> (RootSelection)	802.1W changed the port's role to Root port, using the root selection computation.
Informational	vlan <*vlan-id*> New RootBridge <*mac-address*> RootPort <*portnum*> (BpduRcvd)	802.1W selected a new root bridge as a result of the BPDUs received on a bridge port.

TABLE 222 Brocade **Syslog messages** (Continued)

Message level	Message	Explanation
Informational	vlan <*vlan-id*> Bridge is RootBridge <*mac-address*> (MgmtPriChg)	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.
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)
Informational	Port <*portnum*>, srcip-security max-ipaddr-per-int reached.Last IP=<*ipaddr*>	The address limit specified by the **srcip-security max-ipaddr-per-interface** command has been reached for the port.
Informational	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	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	vlan <*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 222 Brocade Syslog messages (Continued)

Message level	Message	Explanation
Informational	ACL <*acl id*> 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.
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.
Informational	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	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/disable configuration through the Web, SNMP, console, SSH, or Telnet session.
Informational	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.
Informational	Port <*portnum*>, srcip-security max-ipaddr-per-int reached.Last IP=<*ipaddr*>	The address limit specified by the **srcip-security max-ipaddr-per-interface** command has been reached for the port.
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.
Debug	DOT1X: Not enough memory	There is not enough system memory for 802.1x authentication to take place. Contact Brocade Technical Support.

TABLE 223 BFD Syslog messages

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.
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.

Appendix

Software Specifications

B

This appendix lists the following information for the BigIron RX:

- IEEE compliance
- RFC support
- Internet draft support

IEEE compliance

- 802.3ae —10-Gigabit Ethernet
- 802.3x — Flow Control
- 802.3ad — Link Aggregation
- 802.1Q — Virtual Bridged LANs
- 802.1D — MAC Bridges
- 802.1w — Rapid STP
- 802.1s — Multiple Spanning Trees
- 802.1X — User authentication
- 802.3 — Ethernet Like MIB
- Repeater MIB
- Ethernet Interface MIB
- SNMP v1, v2c and V3
- SNMP MIB II

RFC compliance

RFC compliance - BGPv4

- 4271 — BGPv4
- 1745 — OSPF Interactions
- 1997 — Communities & Attributes
- 2439 — Route Flap Dampening
- 2796 — Route Reflection
- 3065 — BGP4 Confederations
- 2842 — Capability Advertisement

- 2918 — Route Refresh Capability
- 1269 — Managed Objects for BGP
- 1657 — Managed Objects for BGP-4 using SMIv2
- 3392 — Capabilities Advertisement with BGP-4
- 2385 — BGP Session Protection through TCP MD5
- 3682 — Generalized TTL Security Mechanism, for eBGP Session Protection

RFC compliance - OSPF

- 2178 — OSPF
- 1583 — OSPF v2
- 3103 — OSPF NSSA
- 1745 — OSPF Interactions
- 1765 — OSPF Database Overflow
- 1850 — OSPF Traps
- 2328 — OSPF v2
- 1850 — OSPF v2 MIB
- 2370 — OSPF Opaque LSA Option
- 3623 — Graceful OSPF Restart

RFC compliance - IS-IS

- 1195 — Routing in TCP/IP and Dual Environments
- 2763 — Dynamic Host Name Exchange
- 2966 — Domain-wide Prefix Distribution
- 3567 — IS-IS Cryptographic Authentication (MD-5)

RFC compliance - RIP

- 1058 — RIP v1
- 1723 — RIP v2
- 1812 — RIP Requirements

RFC compliance - IP Multicast

- 1122 — Host Extensions
- 1256 — ICMP Router Discovery Protocol
- 1112 — IGMP
- 2236 — IGMP v2

- 3376 — IGMP v3
- 2362 — PIM-SM
- 3618 — MSDP
- PIM-DM v1
- 3973 — PIM-DM
- 3446 — Anycast RP
- 1075 — DVMRP v2
- 4541 — Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches
- DVMRP v3-07
- 2283 — MBGP

RFC compliance - general protocols

- 791 — IP
- 792 — ICMP
- 793 — TCP
- 783 — TFTP
- 826 — ARP
- 768 — UDP
- 894 — IP over Ethernet
- 903 — RARP
- 906 — TFTP Bootstrap
- 1027 — Proxy ARP
- 950 — Subnets
- 951 — BootP
- 1122 — Host Extensions for IP Multicasting
- 1256 — IRDP
- 1519 — CIDR
- 1542 — BootP Extensions
- 1812 — Requirements for IPv4 Routers
- 1541 and 1542 — DHCP
- 2131 — BootP/DHCP Helper
- 2768 — VRRP
- 1591 — DNS (client)
- 2578 — Structure of Management Information Version 2 (SMIv2)
- 2579 — Textual Conventions for SMIv2
- 1354 — IP Forwarding Table MIB
- 2784 — Generic Routing Encapsulation (GRE)

- 1305 — Network Time Protocol (Version 3) Specification, Implementation and Analysis
- 1191 — Path MTU Discovery
- 896 — Congestion Control
- 3635 — Pause Control
- 1858 — IP Fragment Filtering
- 1340 — Assigned Numbers

RFC compliance - management

- 1757 — RMON Groups Partial 1, full for 2, 3, 9
- 2068 — HTTP
- 2030 — SNTP
- 2865 — RADIUS
- 2866 — RADIUS Accounting
- 2868 — RADIUS Attributes for Tunnel Protocol
- 2869 — RADIUS Extensions
- 3176 — sFlow
- 2578 — SNMPV2
- 2579 — Textual Conventions for SMIv2
- 3410 — SNMPV3
- 3411— Architecture for SNMP
- 3412 — Message Processing and Dispatching for SNMP
- 3413 — Simple Network Management Protocol (SNMP) Applications
- 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
- 3584 — Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework
- 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

RFC compliance - IPv6 core

- 2373 — IPv6 Addressing Architecture
- 1886 — DNS Extensions to Support IP Version 6
- 1887 — IPv6 Unicast Address Allocation Architecture

- 2374 — IPv6 Aggregatable Global Unicast Address Format
- 2450 — Proposed TLA and NLA Assignment Rules
- 2471 — IPv6 Testing Address Allocation
- 2526 — Reserved IPv6 Subnet Anycast Address
- 2928 — Intial IPv6 subTLA ID Assignments
- 2460 — IPv6 Specification
- 2461 — IPv6 Neighbor Discovery
- 2462 — IPv6 Stateless Address Auto-configuration
- 4443 — ICMPv6
- 3513 — IPv6 Addressing Architecture
- 1981 — IPv6 Path MTU Discovery
- 3587 — IPv6 Global Unicast Address Format
- 2375 — IPv6 Multicast Address Assignments
- 2464 — Transmission of IPv6 over Ethernet Networks
- 2711 — IPv6 Router Alert Option
- 3596 — DNS support

RFC compliance - IPv6 routing

- 2080 — RIPng for IPv6
- 2740 — OSPFv3 for IPv6
- 2545 — Use of MP-BGP-4 for IPv6

RFC compliance - IPv6 multicast

- 3810 — Multicast Listener Discovery Version 2 for IPv6
- 4601 — PIM-SM Protocol Specification
- 2362 — PIM-SM
- 2710 — Multicast Listener Discovery (MLD) for IPv6
- 3306 — Unicast-Prefix-based IPv6 Multicast Addresses

RFC compliance - IPv6 transitioning

- 2893 — Transition Mechanisms for IPv6 Hosts and Routers
- 3056 — Connection of IPv6 Domains through IPv4 Clouds

RFC compliance - IPv6 management

- 2452 — IPv6 MIB for TCP

- 2454 — IPv6 MIB for UDP
- 2465 — IPv6 MIB for Textual Conventions and General Group
- 2466 — IPv6 MIB for ICMPv6 Group

Internet drafts

In addition to the RFCs listed in "RFC compliance", the BigIron RX supports the following Internet drafts:

- Draft-ietf-tcpm-tcpsecure-TCP Security
- IETF Draft_ietf_isis_IPv6 for IS-IS for IPv6
- IETF Draft-vida-mld-v2
- Draft-ietf-idr-restart - Graceful Restart Mechanism for BGP
- Draft-ietf-idr-route-filter
- Draft-holbrook-idmr-igmpv3-ssm - IGMPv3 & MLDv2 for SSM
- Draft-ietf-ssm-arch SSM for IP

Appendix

NIAP-CCEVS Certification

C

Some Brocade devices have passed the Common Criteria (CC) certification testing. This testing is sponsored by the National Information Assurance Partnership (NIAP) - Common Criteria Evaluation and Validation Scheme (CCEVS). For more information regarding the NIAP-CCEVS certification process refer to the following link: *http://www.niap-ccevs.org/*.

In an effort to maintain a proper level of security as it relates to access to network infrastructure resources, Brocade recommends that all Brocade hardware be installed within a secure location that is accessible by approved personnel only.

NIAP-CCEVS certified Brocade equipment and Ironware releases

The following Brocade devices have been NIAP-CCEVS certified. The following IronWare software release must be used to remain compliant with this certification.

TABLE 224 NIAP-CCEVS certified Brocade equipment and IronWare software releases

Brocade product	Brocade IronWare software version	Discussed in
NetIron XMR Family	3.8.00a	*NetIron XMR and NetIron MLX Series Configuration Guide*
NetIron MLX Family	3.8.00a	*NetIron XMR and NetIron MLX Series Configuration Guide*
BigIron RX Family	2.5.00b	*BigIron RX Series Configuration Guide*
FastIron SuperX/SX Family	4.1.00	*FastIron Configuration Guide*
FastIron Edge X Family	4.1.00	*FastIron Configuration Guide*
FastIron GS/LS Family	4.2.00a	*FastIron Configuration Guide*
FastIron Edge Switch Family	4.0.00a	*Security Guide*

Web management access to NIAP-CCEVS certified *Security Guide* equipment

All Brocade devices that are to remain in compliancy with the NIAP-CCEVS certification must disable all remote access through the integrated Web management graphical user interface (GUI). In accordance with NIAP-CCEVS this functionality is considered a security risk and must be disabled.

Please refer to the Brocade Configuration Guides associated with each product in the table "NIAP-CCEVS certified Brocade equipment and IronWare software releases"for detailed instructions on how to disable the Web Management Interface feature.

Local user password changes

Please note that if existing usernames and passwords have been configured on a Brocade device with specific privilege levels (super-user, read-only, port-config) and if you attempt to change a user's password by executing the following syntax.

```
Foundry-Router(config)# user fdryreadonly password <value>
```

The privilege level of this particular user will be changed from its current value to "super-user". The “super-user” level username and password combination provides full access to the Brocade command line interface (CLI). To prevent this from occurring, use the following syntax.

```
Foundry-Router(config)# user fdryreadonly privilege <value> password <value>
```

Appendix

Commands That Require a Reload

D

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 225 lists the 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 cold start. To perform a cold start, do one of the following:

- Enter the **reload** command at the Privileged EXEC level of the CLI.
- Cycle the power by powering down the device, then powering it on again.

NOTE
The **boot system** command does not perform a cold start. It performs a warm start.

TABLE 225 Commands that require a software reload

Command	See ...
max-frame-size	"Setting maximum frame size per PPCR" on page 172
system-max	"Displaying and modifying system parameter default settings" on page 127

Appendix

Index to the CLI Commands

E

This appendix lists the CLI commands discussed in this configuration guide. Look for the CLI command alphabetically by feature. You can also use your browser's search function to find the command you want. When you find the command, click on the link to display the section that discusses that command.

ACLs (IP)

Numbered ACL

Commands	See ...
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*>] [match-all <*tcp-flags*>] [match-any <*tcp-flags*>] [<*icmp-type*>] [established] [precedence <*name*> | <*num*>] [tos <*number*>] [dscp-matching <*number*>] [802.1p-priority-matching <*number*>] [dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>] | [dscp-marking <*number*> dscp-cos-mapping] | [dscp-cos-mapping] [fragment] [non-fragment] [first-fragment] [fragment-offset <*number*>] [spi <*00000000 - ffffffff*>] [log]	"Configuring extended numbered ACLs" on page 520 "Enabling ACL filtering of fragmented or non-fragmented packets" on page 557
access-list <*num*> deny | permit host <*ip-protocol*> any any [log]	
access-list <*num*> deny | permit icmp any any [log] <*icmp-type*> | <*type-number*> <*code-number*>	"ICMP filtering for extended ACLs" on page 558
access-list <*num*> deny | permit any I <*source-ip*> |<*source-ip*>/<*mask-bits*> | <*hostname*> <*wildcard*> [log]	"Configuring standard numbered ACLs" on page 518
access-list <*acl-num*> remark [<*comment-text*>]	"Numbered ACLs: adding a comment" on page 547 "Numbered ACLs: deleting a comment" on page 548
no access-list <*acl-number*> <*entire-deny-or-permit-statement*>	"Deleting ACL entries" on page 549
show access-list <*acl-num*> | all	"Displaying ACL definitions" on page 533

Named ACL

Commands	See ...
ip access-list extended I standard <*acl-name*>	"Named ACLs: adding a comment to a new ACL" on page 548 "Deleting ACL entries" on page 549
ip access-list extended <*string*> | <*num*> deny | permit <*ip-protocol*> <*source-ip*> | <*hostname*> <*wildcard*> [<*operator*> <*source-tcp/udp-port*>] <*destination-ip*> | <*hostname*> <*wildcard*> [<*operator*> <*destination-tcp/udp-port*>] [match-all <*tcp-flags*>] [match-any <*tcp-flags*>] [<*icmp-type*>] [established] [precedence <*name*> | <*num*>] [tos <*number*>] [dscp-matching <*number*>] [802.1p-priority-matching <*number*>] [dscp-marking <*number*> 802.1p-priority-marking <*number*> internal-priority-marking <*number*>] [dscp-marking <*number*> dscp-cos-mapping] [dscp-cos-mapping] [fragment] [non-fragment] [first-fragment] [fragment-offset <*number*>] [spi <*00000000 - ffffffff*>] [log]	"Configuring standard or extended named ACLs" on page 529 "Enabling ACL filtering of fragmented or non-fragmented packets" on page 557
ip access-list extended <*string*> I <*num*> deny | permit host <*ip-protocol*> any any [log]	
ip access-list extended <*acl-name*> deny | permit host icmp any any [log] <*icmp-type*> | <*type-number*> <*code-number*>	"ICMP filtering for extended ACLs" on page 558
ip access-list standard <*string*> deny | permit <*source-ip*> | <*hostname*> <*wildcard*> [log]	"Configuring standard or extended named ACLs" on page 529
ip access-list standard <*string*> deny | permit <*source-ip*>/<*mask-bits*> | <*hostname*> [log]	
ip access-list standard <*string*> deny | permit any [log]	
ip access-list standard <*string*> deny | permit host <*source-ip*> | <*hostname*> [log]	
no <*entire-deny-or-permit-statement*>	"From named ACLs" on page 550
remark <*string*>	"Named ACLs: adding a comment to a new ACL" on page 548 "Named ACLs: deleting a comment" on page 549
show access-list name <*acl-name*>	"Displaying ACL definitions" on page 533

Other ACL commands

Commands	See ...
acl-duplication-check	"Enabling ACL duplication check" on page 554
clear access-list all | ethernet <*slot*>/<*port*> | ve <*ve-num*>	"Clearing the ACL statistics" on page 556

Commands	See ...
ip access-group <*num*> I <*name*> in	"Configuring standard numbered ACLs" on page 518 "Configuring extended numbered ACLs" on page 520 "Configuring standard or extended named ACLs" on page 529
ip access-group <*num*> in ethernet <*portnum*> [<*portnum*>...] to <*portnum*>	"Applying ACLs to a virtual routing interface" on page 551
ip access-list logging-age <*minutes*>	"Configuring the Layer 4 session log timer" on page 552
ip rebind-acl <*num*> | <*name*> | all	"Reapplying modified ACLs" on page 551
ip show-acl-service-number	"Displaying of TCP/UDP numbers in ACLs" on page 534
show access-list accounting brief	"Displaying accounting statistics for all ACLs" on page 555
show access-list accounting ethernet [<*slot*>/<*port*> | ve <*ve-number*>]	"Displaying statistics for an interface" on page 555
system-max ip-filter-sys <*num*>	"Enabling support for additional ACL statements" on page 514

ACLs (L2)

Commands	See ...
access-list <*num*> permit | deny <*src-mac*> <*mask*> | any <*dest-mac*> <*mask*> | any [<*vlan-id*> | any [etype <*etype-str*>] [log-enable]]	"Creating a Layer 2 ACL table" on page 506
mac access-group <*num*> in	"Binding a Layer 2 ACL table to an interface" on page 508
show access-list <*num*>	"Viewing Layer 2 ACLs" on page 508
system-max l2-acl-table-entries <*max*>	"Increasing the maximum number of clauses per Layer 2 ACL table" on page 508

BGP4

Commands	See ...
address-family ipv4 unicast | ipv4 multicast	"Entering and exiting the address family configuration level" on page 749
address-filter <*num*> permit | deny <*ip-addr*> <*wildcard*> <*mask*> <*wildcard*>	"Filtering specific IP addresses" on page 749
aggregate-address <*ip-addr*> <*ip-mask*> [as-set] [summary-only] [suppress-map <*map-name*>] [advertise-map <*map-name*>] [attribute-map <*map-name*>]	"Aggregating routes advertised to BGP4 neighbors" on page 757

Commands	See ...
always-compare-med	"Configuring the BigIron RX to always compare Multi-Exit Discriminators (MEDs)" on page 757
as-path-filter <*num*> permit | deny <*as-path*>	"Defining an AS-path filter" on page 751
as-path-ignore	"Disabling or re-enabling comparison of the AS-path length" on page 758
clear ip bgp damping [<*ip-addr*> <*ip-mask*>]	"Removing route dampening from a route" on page 812 "Removing route flap dampening" on page 821
clear ip bgp flap-statistics [regular-expression <*regular-expression*> | <*address*> <*mask*> | neighbor <*ip-addr*>]	"Clearing route flap dampening statistics" on page 814 "Clearing route flap dampening statistics" on page 821
clear ip bgp neighbor <*ip-addr*> [soft in prefix-filter]	"Sending and receiving ORFs" on page 808
clear ip bgp neighbor <*ip-addr*> | <*peer-group-name*> soft in	"Placing a policy change into effect" on page 815
clear ip bgp neighbor all | <*ip-addr*> | <*peer-group-name*> | <*as-num*> [soft-outbound | soft [in | out]]	"Dynamically refreshing routes" on page 818
clear ip bgp neighbor all | <*ip-addr*> | <*peer-group-name*> | <*as-num*> traffic	"Clearing traffic counters" on page 820
clear ip bgp neighbor all | <*ip-addr*> | <*peer-group-name*> | <*as-num*> last-packet-with-error | notification-errors	"Clearing diagnostic buffers" on page 822
clear ip bgp routes [<*ip-addr*>/<*prefix-length*>]	"Clearing and resetting BGP4 routes in the IP route table" on page 820
clear ip bgp traffic	"Clearing traffic counters" on page 820
client-to-client-reflection	"Disabling or re-enabling client-to-client route reflection" on page 759
cluster-id <*num*> | <*ip-addr*>	"Configuring a route reflector" on page 759 "Configuration procedures" on page 792
community-filter <*num*> permit | deny <*num*>:<*num*> | internet | local-as | no-advertise | no-export	"Defining a community filter" on page 751
compare-routerid	"Enabling or disabling comparison of the router IDs" on page 759
always-compare-med	"Configuring the BigIron RX to always compare Multi-Exit Discriminators (MEDs)" on page 757
confederation identifier <*num*>	"Configuring a BGP confederation" on page 761
confederation peers <*num*> [<*num*> ...]	"Configuring a BGP confederation" on page 761
dampening [<*half-life*> <*reuse*> <*suppress*> <*max-suppress-time*>]	"Configuring route flap dampening" on page 763
default-information-originate	"Originating the default route" on page 764
default-local-preference <*num*>	"Changing the default local preference" on page 764
default-metric <*num*>	"Changing the default metric used for redistribution" on page 765

Commands	See ...
distance <*external-distance*> <*internal-distance*> <*local-distance*>	"Changing administrative distances" on page 765
enforce-first-as	"Requiring the first AS to be the neighbor's AS" on page 766
exit-address-family	"Entering and exiting the address family configuration level" on page 749
fast-external-fallover	"Enabling fast external fallover" on page 767
interface loopback <*num*>	"Adding a loopback interface" on page 789
ip as-path access-list <*string*> [seq <*seq-value*>] deny | permit <*regular-expression*>	"Defining an AS-path ACL" on page 793
ip community-list extended <*string*> [seq <*seq-value*>] deny | permit <*community-num*> | <*regular-expression*>	"Defining a community ACL" on page 796
ip community-list standard <*string*> [seq <*seq-value*>] deny | permit <*community-num*>	"Defining a community ACL" on page 796
ip prefix-list <*name*> [seq <*seq-value*>] [description <*string*>] deny | permit <*network-addr*>/<*mask-bits*> [ge <*ge-value*>] [le <*le-value*>]	"Defining and applying IP prefix lists" on page 797
ip router-id <*ip-addr*>	"Changing the router ID" on page 788
local-as <*num*>	"Configuring a BGP confederation" on page 761 "Setting the local AS number" on page 767
match [as-path <*name*>] | [address-filters | as-path-filters | community-filters <*num,num,...*>] | [community <*acl*> exact-match] | [ip address <*acl*> | prefix-list <*string*>] | [ip route-source <*acl*> | prefix <*name*>] [metric <*num*>] | [next-hop <*address-filter-list*>] | [level-1 | level-2 | level-1-2] [route-type internal | external-type1 | external-type2] | [tag <*tag-value*>]	"Specifying the match conditions" on page 800
match as-path <*num*>	"Matching based on AS-path ACL" on page 802
match community <*acl*> exact-match	"Matching on routes containing a specific set of communities" on page 803
match community <*string*>	"Matching based on community ACL" on page 802
match ip address <*name-or-num*>	"Matching based on destination network" on page 802
match ip address prefix-list <*name*>	"Matching based on destination network" on page 802
match ip next-hop <*num*>	"Matching based on next-hop router" on page 803
match ip next-hop prefix-list <*name*>	"Matching based on next-hop router" on page 803
match ip route-source <*acl*> | prefix <*name*>	"Matching based on the route source" on page 803
maximum-paths <*num*>	"Changing the maximum number of shared BGP4 paths" on page 768

Commands	See ...
med-missing-as-worst	"Treating missing MEDs as the worst MEDs" on page 768
multipath ebgp | ibgp | multi-as	"Customizing BGP4 load sharing" on page 769
neighbor <*ip-addr*> |<*peer-group-name*> password [0 | 1] <*string*>	"Encryption example" on page 775
neighbor <*ip-addr*> | <*peer-group-name*> capability orf prefixlist [send | receive]	"Enabling cooperative filtering" on page 807
neighbor <*ip-addr*> | <*peer-group-name*> soft-reconfiguration inbound	"Enabling soft reconfiguration" on page 815
neighbor <*ip-addr*> | <*peer-group-name*> unsuppress-map <*map-name*>	"Removing route dampening from suppressed neighbor's routes" on page 773
neighbor <*ip-addr*>|<*peer-group-name*> [advertisement-interval <*num*>] [capability orf prefixlist [send | receive]] [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] [nlri multicast | unicast | multicast unicast] [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] [timers keep-alive <*num*> hold-time <*num*>] [unsuppress-map <*map-name*>] [update-source <*ip-addr*> | ethernet <*portnum*> | loopback <*num*> | ve <*num*>] [weight <*num*>]	"Configuring BGP4 neighbors" on page 769 "Configuring a peer group" on page 777
neighbor <*ip-addr*> distribute-list <*name-or-num*> in | out	"Defining neighbor distribute lists" on page 798
neighbor <*ip-addr*> peer-group <*peer-group-name*>	"Applying a peer group to a neighbor" on page 778
neighbor <*ip-addr*> route-reflector-client	"Configuration procedures" on page 792
neighbor <*ip-addr*> shutdown	"Administratively shutting down a session with a BGP4 neighbor" on page 779
neighbor <*peer-group-name*> peer-group	"Configuring a peer group" on page 777
network <*ip-addr*> <*ip-mask*> [route-map <*map-name*>] | [weight <*num*>] | [backdoor]	"Specifying a list of networks to advertise" on page 779 "Specifying a route map name when configuring BGP4 network information" on page 780
next-hop-enable-default	"Using the IP default route as a valid next hop for a BGP4 route" on page 781

Commands	See ...
next-hop-recursion	"Enabling recursive next-hop lookups" on page 784
redistribute connected [metric <*num*>] [route-map <*map-name*>]	"Redistributing connected routes" on page 785
redistribute connected | ospf | rip | isis | static	"Using the IP default route as a valid next hop for a BGP4 route" on page 781
redistribute isis level-1 | level-1-2 | level-2 [metric <*num*>] [route-map <*map-name*>]	"Redistributing ISIS" on page 786
redistribute ospf [match internal | external1 | external2] [metric <*num*>] [route-map <*map-name*>]	"Redistributing OSPF external routes" on page 785
redistribute rip [metric <*num*>] [route-map <*map-name*>]	"Redistributing RIP routes" on page 785
redistribute static [metric <*num*>] [route-map <*map-name*>]	"Redistributing static routes" on page 786
route-map <*map-name*> permit | deny <*num*>	"Entering the route map Into the software" on page 800
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 <*tag-value*>] | [weight <*num*>]	"Setting parameters in the routes" on page 804
set comm-list <*acl*> delete	"Deleting a community from a BGP4 route" on page 806
set ip next-hop peer-address	"Setting the next hop of a BGP4 route" on page 806
set metric-type internal	"Setting a BP4 route's MED to be equal to the next-hop route IGP metric" on page 805
show ip bgp [route] <*ip-addr*>/<*prefix*> [longer-prefixes] | <*ip-addr*>	"Displaying information for a specific route" on page 842
show ip bgp attribute-entries	"Displaying BGP4 route-attribute entries" on page 846
show ip bgp config	"Displaying the active BGP4 configuration" on page 825
show ip bgp filtered-routes [<*ip-addr*>] | [as-path-access-list <*num*>] | [detail] | [prefix-list <*string*>]	"Displaying the filtered routes received from the neighbor or peer group" on page 816

Commands	See ...
show ip bgp flap-statistics [regular-expression <*regular-expression*> | <*address*> <*mask*> [longer-prefixes] | neighbor <*ip-addr*> | filter-list <*num*>...]	"Displaying route flap dampening statistics" on page 848 "Clearing route flap dampening statistics" on page 814
show ip bgp neighbor <*ip-addr*>	"Displaying cooperative filtering information" on page 809
show ip bgp neighbor <*ip-addr*> advertised-routes [<*ip-addr*>/<*prefix*>]	"Displaying advertised routes" on page 837
show ip bgp neighbor <*ip-addr*> received prefix-filter	"Displaying cooperative filtering information" on page 809
show ip bgp neighbor <*ip-addr*> rib-out-routes [<*ip-addr*>/<*prefix*>]	"Displaying the adj-RIB-out for a neighbor" on page 838
show ip bgp neighbor <*ip-addr*> routes unreachable	"Displaying the routes whose destinations are unreachable" on page 837
show ip bgp neighbors <*ip-addr*> received-routes [detail]	"Displaying all the routes received from the neighbor" on page 817
show ip bgp neighbors [<*ip-addr*>] | [route-summary]	"Displaying summary neighbor information" on page 826
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]]	"Displaying BGP4 neighbor information" on page 827
show ip bgp peer-group [<*peer-group-name*>]	"Displaying peer group information" on page 838
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]	"Displaying the BGP4 route table" on page 839
show ip bgp routes best	"Displaying the best BGP4 routes" on page 841
show ip bgp routes detail	"Displaying route details" on page 844
show ip bgp routes summary	"Displaying summary route information" on page 838
show ip bgp routes unreachable	"Displaying BGP4 routes whose destinations are unreachable" on page 841
show ip bgp summary	"Displaying summary BGP4 information" on page 823
show ip route [<*ip-addr*> | <*num*> | bgp | ospf | rip | isis]	"Displaying the routes BGP4 has placed in the IP route table" on page 847
show route-map [<*map-name*>]	"Displaying the active route map configuration" on page 849

Commands	See ...
snmp-server enable traps bgp	"Generating traps for BGP" on page 814
timers keep-alive <*num*> hold-time <*num*>	"Changing the keep alive time and hold time" on page 787
update-time <*secs*>	"Changing the BGP4 next-hop update timer" on page 788

FDP/CDP

Commands	See ...
cdp enable	"Enabling interception of CDP packets on an interface" on page 1016
cdp run	"Enabling interception of CDP packets globally" on page 1016
clear fdp counters	"Clearing FDP and CDP statistics" on page 1015 "Clearing CDP information" on page 1018
clear fdp table	"Clearing FDP and CDP neighbor information" on page 1015
fdp enable	"Enabling FDP at the interface level" on page 1012
fdp holdtime <*secs*>	"Changing the FDP hold time" on page 1012
fdp run	"Enabling FDP globally" on page 1011
fdp timer <*secs*>	"Changing the FDP update timer" on page 1012
show fdp entry * | <*device-id*>	"Displaying FDP entries" on page 1014 "Displaying CDP entries" on page 1017
show fdp interface [ethernet <*slot*>/<*portnum*>]	"Displaying FDP information for an interface" on page 1014
show fdp neighbor [ethernet<*slot*>/<*portnum*>] [detail]	"Displaying neighbor information" on page 1013
show fdp neighbors [detail | ethernet <*portnum*>]	"Displaying neighbors" on page 1016
show fdp traffic	"Displaying FDP and CDP statistics" on page 1015 "Displaying CDP statistics" on page 1018

IP

Commands	See ...
arp<*ip-addr*> <*mac-addr*> ethernet <*slot/port*>	"Creating static ARP entries" on page 183
bootp-relay-max-hops <*1-15*>	"Changing the maximum number of hops to a BootP relay server" on page 212
clear ip route [<*ip-addr*> <*ip-mask*> | <*ip-addr*>/<*mask-bits*>]	"Clearing IP routes" on page 223
default-mtu <*num*>	"Globally changing the IP MTU" on page 174

Commands	See ...
interface ethernet <*slot/port*> I loopback <*num*>	"Assigning an IP address to an Ethernet port" on page 154 "Assigning an IP address to a loopback interface" on page 155
interface ve <*num*>	"Assigning an IP address to a virtual interface" on page 155
ip address <*ip-addr*> <*ip-mask*> | <*ip-addr*>/<*mask-bits*> [ospf-ignore I ospf-passive I secondary]	"Assigning an IP address to an Ethernet port" on page 154 "Assigning an IP address to a loopback interface" on page 155 "Assigning an IP address to a virtual interface" on page 155 "Configuring the default gateway" on page 157 "Deleting an IP address" on page 156
ip arp-age <*num*>	"Changing the ARP aging period" on page 182
ip bootp-gateway <*ip-addr*>	"ip bootp-gateway <ip-addr>" on page 212
ip broadcast-zero	"Enabling support for zero-based IP subnet broadcasts" on page 188
ip default-gateway <*ip-addr*><*ip-mask*> I <*ip-addr*>/<*mask-bits*>	"Configuring the default gateway" on page 157
ip default-network <*ip-addr*>	"Configuring a default network route" on page 201
ip directed-broadcast	"Enabling forwarding of directed broadcasts" on page 187
ip dns domain-name <*name*>	"Defining a DNS entry" on page 166
ip dns server-address <*ip-addr*> [<*ip-addr*>] [<*ip-addr*>] [<*ip-addr*>]	"Defining a DNS entry" on page 166
ip dr-aggregate	"Dropping traffic sent to the null0 interface in hardware" on page 195
ip encapsulation snap | ethernet-2	"Changing the encapsulation type" on page 171
ip forward-protocol udp <*udp-port-name*> | <*udp-port-num*>	"Enabling forwarding for a UDP application" on page 209
ip helper-address <*ip-addr*>	"Configuring an IP helper address" on page 212
ip hw-drop-on-def-route	"Dropping traffic sent to the null0 interface in hardware" on page 195
ip icmp echo broadcast-request	"Disabling replies to broadcast ping requests" on page 189
ip icmp redirects	"Disabling ICMP redirect messages" on page 190
ip icmp unreachable [network | host | protocol | administration | fragmentation-needed | port | source-route-fail]	"Disabling ICMP destination unreachable messages" on page 189
ip irdp [broadcast | multicast] [holdtime <*seconds*>] [maxadvertinterval <*seconds*>] [minadvertinterval <*seconds*>] [preference <*number*>]	"Enabling IRDP globally" on page 207 "Enabling IRDP on an individual port" on page 207
ip load-sharing [<*number*>]	"Changing the maximum number of load sharing paths" on page 204

Commands	See ...
ip mtu <*num*>	"Changing the maximum transmission unit on an individual interface" on page 174
ip proxy-arp	"Enabling proxy ARP" on page 182
ip redirect	"Disabling ICMP redirect messages" on page 190
ip route <*dest-ip-addr*> <*dest-mask*> | <*dest-ip-addr*>/<*mask-bits*> <next-hop-ip-addr> | ethernet <*slot/port*> | ve <*num*> [<*metric*>] [tag <*num*>] [distance <*num*>]	"Configuring a static IP route" on page 193
ip route <*ip-addr*> <*ip-mask*> | <*dest-ip-addr*>/<*mask-bits*> null0 [<*metric*>] [tag <*num*>] [distance <*num*>]	"Configuring a "null" route" on page 194
ip router-id <*ip-addr*>	"Changing the router ID" on page 174
Ip show-portname	"Displaying interface name in Syslog" on page 216
ip show-subnet-length	"Changing the network mask display to prefix format" on page 156
ip source-route	"Disabling forwarding of IP source-routed packets" on page 187
ip radius source-interface ethernet <*slot/port*> | loopback <*num*> | ve <*num*>	"Specifying a single source interface for Telnet, TACACS/TACACS+, or RADIUS packets" on page 175
ip tacacs source-interface ethernet <*slot/port*> | loopback <*num*> | ve <*num*>	"Specifying a single source interface for Telnet, TACACS/TACACS+, or RADIUS packets" on page 175
ip telnet source-interface ethernet<*slot/port*> | loopback <*num*> | ve <*num*>	"Specifying a single source interface for Telnet, TACACS/TACACS+, or RADIUS packets" on page 175
ip ttl <*1-255*>	"Changing the TTL threshold" on page 186
rate-limit-arp <*num*>	"Rate limiting ARP packets" on page 180
router-interface ve <*num*>	"Assigning an IP address to a virtual interface" on page 155
show arp [ethernet <*slot/port*> | mac-address <xxxx.xxxx.xxxx> [<*mask*>] | <*ip-addr*> [<*ip-mask*>]] [<*num*>] [| begin<*expression*> | exclude<*expression*> | include<*expression*>]	"Displaying the ARP cache" on page 217
show ip	"Displaying global IP configuration information" on page 213
show ip cache [<*ip-addr*>] [| begin<*expression*> | exclude<*expression*> | include<*expression*>]	"Displaying the forwarding cache" on page 219
show ip interface [ethernet <*slot/port*>] | [loopback <*num*>] | [ve <*num*>]	"Displaying IP interface information" on page 215
show ip route <*num*> | [<*ip-addr*> [<*ip-mask*>] [debug | detail | longer]] | connected | bgp | isis | ospf | rip | static | summary] [| begin<*expression*> | exclude<*expression*> | include<*expression*>]	"Displaying the IP route table" on page 220

Commands	See ...
show ip route summary	"Displaying the IP route table" on page 220
show ip static-arp [ethernet <*portnum*> | mac-address <xxxx.xxxx.xxxx> [<*mask*>] | <*ip-addr*> [<*ip-mask*>]] [<*num*>] [| begin<*expression*> | exclude<*expression*> | include<*expression*>]	"Displaying the static ARP table" on page 218
show ip traffic	"Displaying IP traffic statistics" on page 223
system-max ip-static-arp <*num*>	"Changing the maximum number of entries the static ARP table can hold" on page 184
traceroute <*host-ip-addr*> [maxttl <*value*>] [minttl <*value*>] [numeric] [timeout <*value*>] [source-ip <*ip addr*>]	"Using a DNS name to initiate a trace route" on page 170

Metro Ring protocol

Commands	See ...
metro-ring <*ring-id*>	"Configuring MRP with shared interfaces" on page 403
name <*string*>	"Configuring MRP with shared interfaces" on page 403
master	"Adding an MRP ring to a VLAN" on page 396
ring-interface ethernet <*primary-if*> ethernet <*secondary-if*>	"Adding an MRP ring to a VLAN" on page 396
enable	"Configuring MRP with shared interfaces" on page 403
hello-time <*ms*>	"Changing the hello and preforwarding times" on page 397
preforwarding-time <*ms*>	"Changing the hello and preforwarding times" on page 397
diagnostics	"Enabling MRP diagnostics" on page 404
show metro <*ring-id*> diag	"Displaying MRP diagnostics" on page 404 "Displaying ring information" on page 406
show topology-group [<*group-id*>]	"Displaying topology group information" on page 405
show metro [<*ring-id*>]	"Displaying ring information" on page 406

IPv6 BGP4+

Commands	See ...
aggregate-address <*ipv6-prefix*>/<*prefix-length*> [as-set] [summary-only] [suppress-map <*map-name*>] [advertise-map <*map-name*>] [attribute-map <*map-name*>]	"Aggregating routes advertised to BGP4 neighbors" on page 1127
clear ip bgp routes [<*ipv6-prefix*>/<*prefix-length*>]	"Clearing and resetting BGP4+ routes in the IPv6 route table" on page 1132
clear ipv6 bgp dampening [<*ipv6-prefix*>/<*prefix-length*>]	"Removing route flap dampening" on page 1128
clear ipv6 bgp flap-statistics [<*ipv6-prefix*>/<*prefix-length*> | neighbor <*ipv6-address*> | regular-expression <*regular-expression*>]	"Clearing route flap dampening statistics" on page 1129
clear ipv6 bgp local routes	"Clearing BGP4+ neighbor diagnostic buffers" on page 1130
clear ipv6 bgp neighbor <*ipv6-address*> | <*peer-group-name*> [soft in prefix-filter]	"Resetting a BGP4+ neighbor session to send and receive ORFs" on page 1130
clear ipv6 bgp neighbor <*ipv6-address*> flap-statistics	"Clearing BGP4+ neighbor route flap dampening statistics" on page 1132
clear ipv6 bgp neighbor all | <*ipv6-address*> | <*peer-group-name*> | <*as-number*> [soft-outbound | soft [in | out]]	"Clearing and resetting BGP4+ routes in the IPv6 route table" on page 1132
clear ipv6 bgp neighbor all | <*ipv6-address*> | <*peer-group-name*> | <*as-number*> traffic	"Clearing BGP4+ neighbor diagnostic buffers" on page 1130
clear ipv6 bgp neighbor all | <*ipv6-address*> | <*peer-group-name*> | <*as-number*> last-packet-with-error | notification-errors	"Clearing BGP4+ neighbor diagnostic buffers" on page 1130
clear ipv6 bgp traffic	"Clearing BGP4+ neighbor traffic counters" on page 1132
default-information-originate	"Advertising the default BGP4+ route" on page 1125
local-as <*number*>	"Enabling BGP4+" on page 1121
match ipv6 address prefix-list <*name*>	"Configuring a route map" on page 1123
neighbor <*ipv6-address*> default-originate [route-map <*name*>]	"Assigning IPv6 neighbor to peer group" on page 1125
neighbor <*ipv6-address*> peer-group <*peer-group-name*>	"Assigning IPv6 neighbor to peer group" on page 1125
neighbor <*ipv6-address*> remote-as <*as-number*>	"Adding a neighbor to a local router" on page 1124
neighbor <*ipv6-address*> remote-as <*as-number*>	"Adding BGP4+ neighbors using link-local addresses" on page 1122
neighbor <*ipv6-address*> remote-as <*as-number*>	"Adding a neighbor to a local router" on page 1124
neighbor <*ipv6-address*> route-map [in | out] <*name*>	"Configuring a route map" on page 1123
neighbor <*ipv6-address*> update-source <*ipv4-address*> | ethernet <*port*> | loopback <*number*> | ve <*number*>	"Identifying a neighbor interface" on page 1123

Commands	See ...
neighbor <*peer-group-name*> peer-group	"Creating a BGP4+ peer group" on page 1124
network <*ipv6-prefix*>/<*prefix-length*> [route-map <*name*>]	"Importing routes into BGP4+" on page 1126
redistribute <*protocol*> [level-1 | level-1-2 | level-2] [match external1 | external2 | internal] [metric <*metric-value*>] [route-map <*name*>]	"Redistributing prefixes into BGP4+" on page 1126
route-map <*name*> deny | permit <*sequence-number*>	"Configuring a route map" on page 1123
router bgp	"Enabling BGP4+" on page 1121
set ipv6 next-hop <*ipv6-address*>	"Configuring a route map" on page 1123
show ipv6 bgp <*ipv6-prefix*>/<*prefix-length*> [longer-prefixes]	"Displaying BGP4+ route information" on page 1139
show ipv6 bgp attribute-entries	"Displaying BGP4+ route-attribute entries" on page 1140
show ipv6 bgp config	"Displaying the BGP4+ running configuration" on page 1142
show ipv6 bgp dampened-paths	"Displaying route flap dampening statistics" on page 1148
show ipv6 bgp filtered-routes [<*ipv6-prefix*>/<*prefix-length*> [longer-prefixes] | [as-path-access-list <*name*>] | [prefix-list <*name*>]	"Displaying filtered-out BGP4+ routes" on page 1143
show ipv6 bgp filtered-routes detail [<*ipv6-prefix*>/<*prefix-length*> [longer-prefixes] | [as-path-access-list <*name*>] | [prefix-list <*name*>]	"Displaying filtered-out BGP4+ routes" on page 1143
show ipv6 bgp flap-statistics [<*ipv6-prefix*>/<*prefix-length*> [longer-prefixes] | as-path-filter <*number*> | neighbor <*ipv6-address*> | regular-expression <*regular-expression*>]	"Displaying route flap dampening statistics" on page 1148
show ipv6 bgp neighbor [<*ipv6-address*>]	"Displaying BGP4+ neighbor information" on page 1149
show ipv6 bgp neighbor [<*ipv6-address*>] routes-summary	"Displaying BGP4+ summary" on page 1172
show ipv6 bgp neighbor <*ipv6-address*> advertised-routes [detail] <*ipv6-prefix*>/<*prefix-length*>	"Displaying routes advertised to a BGP4+ neighbor" on page 1155
show ipv6 bgp neighbor <*ipv6-address*> attribute-entries	"Displaying BGP4+ route-attribute entries" on page 1140
show ipv6 bgp neighbor <*ipv6-address*> flap-statistics	"Displaying route flap dampening statistics" on page 1148
show ipv6 bgp neighbor <*ipv6-address*> received prefix-filter	"Displaying outbound route filters received from a BGP4+ neighbor" on page 1160
show ipv6 bgp neighbor <*ipv6-address*> received-routes [detail]	"Displaying routes received from a BGP4+ neighbor" on page 1160
show ipv6 bgp neighbor <*ipv6-address*> rib-out-routes [<*ipv6-prefix*>/<*prefix-length*> | detail [<*ipv6-prefix*>/<*prefix-length*> <*network-mask*>]]	"Displaying the adj-RIB-out for a BGP4+ neighbor" on page 1164

Commands	See ...
show ipv6 bgp neighbor <*ipv6-address*> routes best | detail [best | unreachable] | unreachable	"Displaying the best and unreachable routes received from a BGP4+ neighbor" on page 1167
show ipv6 bgp neighbor last-packet-with-error	"Displaying IPv6 neighbor route summary information" on page 1169
show ipv6 bgp peer-group [<*peer-group-name*>]	"Displaying BGP4+ peer group configuration information" on page 1171
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]	"Displaying the BGP4+ route table" on page 1133
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]	"Displaying the BGP4+ route table" on page 1133
show ipv6 bgp summary	"Displaying BGP4+ summary" on page 1172

IPv6 ACL

Commands	See ...
ipv6 access-list <*acl name*>	"For ICMP" on page 1190 "For TCP" on page 1190 "For UDP" on page 1190 "For IPv6 and supported protocols other than ICMP, TCP, or UDP" on page 1189
ipv6 access-list <*name*> deny | permit <*ipv6-source-prefix*>/<*prefix-length*> | any <*ipv6-destination-prefix*>/<*prefix-length*> | any [sequence <*number*>]	"For IPv6 and supported protocols other than ICMP, TCP, or UDP" on page 1189
ipv6 access-list <*name*> deny | permit <*ipv6-source-prefix*>/<*prefix-length*> | any <*ipv6-destination-prefix*>/<*prefix-length*> | any [sequence <*number*>] dscp <*dscp-value*>	"Configuring an IPv6 ACL" on page 1186

Commands	See ...
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*>]]	"For TCP" on page 1190
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*>]]	"ACL syntax" on page 1189
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*>]	"For ICMP" on page 1190
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*>]]	"For UDP" on page 1190
remark <*comment-text*>	"Adding a comment to an IPv6 ACL entry" on page 1195
remark-entry sequence <*sequence number*> <*comment-text*>	"Adding a comment to an IPv6 ACL entry" on page 1195
show ipv6 access-list [<*access-list-name*>]	"Adding a comment to an IPv6 ACL entry" on page 1195
show running-config	"Adding a comment to an IPv6 ACL entry" on page 1195

IPv6 basic connectivity

Commands	See ...
ipv6 enable	"Configuring a link-local IPv6 address" on page 1069
ipv6 nd reachable-time <*seconds*>	"Configuring reachable time for remote IPv6 nodes" on page 1089
ipv6 unicast-routing	"Configuring IPv4 and IPv6 protocol stacks" on page 1078
clear ipv6 cache [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | ethernet <*port*> | tunnel <*number*> | ve <*number*>]	"Clearing the IPv6 cache" on page 1092
clear ipv6 flows	"Deleting IPv6 session flows" on page 1094

Commands	See ...
clear ipv6 neighbor [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | ethernet <*port*> | ve <*number*>]	"Clearing IPv6 neighbor information" on page 1093
clear ipv6 route [<*ipv6-prefix*>/<*prefix-length*>]	"Clearing IPv6 routes from the IPv6 route table" on page 1093
clear ipv6 traffic	"Clearing IPv6 traffic statistics" on page 1094
ip address <*ip-address*> <*sub-net-mask*> [secondary]	"Configuring IPv4 and IPv6 protocol stacks" on page 1078
ip dns server-address <*ip-addr*> [<*ip-addr*>] [<*ip-addr*>] [<*ip-addr*>]	"Defining a DNS entry" on page 1079
ipv6 address <*ipv6-address*> link-local	"Configuring a link-local IPv6 address" on page 1069
ipv6 address <*ipv6-prefix*>/<*prefix-length*> [anycast]	"Configuring IPv6 anycast addresses" on page 1070
ipv6 address <*ipv6-prefix*>/<*prefix-length*> [eui-64]	"Configuring IPv4 and IPv6 protocol stacks" on page 1078
ipv6 address <*ipv6-prefix*>/<*prefix-length*>	"Configuring a global or site-local IPv6 address" on page 1068
ipv6 address <*ipv6-prefix*>/<*prefix-length*> eui-64	"Configuring a global or site-local IPv6 address with a manually configured interface ID" on page 1069
ipv6 dns domain-name <*domain name*>	"Defining a DNS entry" on page 1079
ipv6 dns server-address <*ipv6-addr*> [<*ipv6-addr*>] [<*ipv6-addr*>] [<*ipv6-addr*>]	"Defining a DNS entry" on page 1079
ipv6 hop-limit <*number*>	"Limiting the number of hops an IPv6 packet can traverse" on page 1091
ipv6 icmp error-interval <*interval*> [<*number-of-tokens*>]	"Configuring ICMP rate limiting" on page 1083
ipv6 load-sharing [<*num*>]	"Changing the maximum number of load sharing paths for IPv6" on page 1081
ipv6 load-sharing by-host	"ECMP load sharing for IPv6" on page 1080
ipv6 mtu <*bytes*>	"Changing the IPv6 MTU" on page 1090
ipv6 mtu <*value*>	"Changing the IPv6 MTU" on page 1090
ipv6 nd dad attempt <*number*>	"Setting neighbor solicitation parameters for duplicate address detection" on page 1086
ipv6 nd managed-config-flag	"Setting IPv6 router advertisement parameters" on page 1087
ipv6 nd other-config-flag	"Setting flags in IPv6 router advertisement messages" on page 1088
ipv6 nd prefix-advertisement <*ipv6-prefix*>/<*prefix-length*> <*valid-lifetime*> <*preferred-lifetime*> [autoconfig] [onlink]	"Controlling prefixes advertised in IPv6 router advertisement messages" on page 1088
ipv6 nd ra-interval <*number*>	"Setting IPv6 router advertisement parameters" on page 1087
ipv6 nd ra-lifetime <*number*>	"Setting IPv6 router advertisement parameters" on page 1087
ipv6 nd suppress-ra	"Enabling and disabling IPv6 router advertisements" on page 1089

Commands	See ...
ipv6 neighbor <*ipv6-address*> ethernet <*port*> | ve <*ve-number*> [ethernet <*port*>] <*link-layer-address*>	"Configuring static neighbor entries" on page 1091
ipv6 redirects	"Disabling or reenabling ICMP redirect messages" on page 1084
ipv6 unicast-routing	"Enabling IPv6 routing" on page 1068
log host ipv6 <*ipv6-address*> [<*udp-port-num*>]	"Configuring an IPv6 Syslog server" on page 1074
port-priority	"QoS for IPv6 traffic" on page 1091
show ipv6	"Displaying ECMP load-sharing information for IPv6" on page 1082
show ipv6 cache [<*index-number*> | <*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | ethernet <*port*> | ve <*number*> | tunnel <*number*>]	"Displaying ECMP load-sharing information for IPv6" on page 1082
show ipv6 flows [<source-ipv6-prefix/prefix-length> | any | host <*source-ipv6_address*> <destination-ipv6-prefix/prefix-length> | any | host <*destination-ipv6-address*>]	"Displaying IPv6 session flows" on page 1107
show ipv6 interface [<*interface*> [<*port-number*> |<*number*>]]	"Displaying IPv6 interface information" on page 1095
show ipv6 neighbor [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | <*interface*> [<*port*> |<*number*>]]	"Displaying IPv6 neighbor information" on page 1097
show ipv6 route [<*ipv6-address*> | <*ipv6-prefix*>/<*prefix-length*> | bgp | connect | ospf | rip | isis | static | summary]	"Displaying the IPv6 route table" on page 1098
show ipv6 router	"Displaying local IPv6 routers" on page 1099
show ipv6 tcp connections	"Displaying IPv6 TCP information" on page 1100
show ipv6 tcp status <*local-ip-address*> <*local-port-number*> <*remote-ip-address*> <*remote-port-number*>	"Displaying IPv6 TCP information" on page 1100
show ipv6 traffic	"Displaying IPv6 traffic statistics" on page 1104
snmp-client ipv6 <*ipv6-address*>	"Restricting SNMP access to an IPv6 node" on page 1073
snmp-server host ipv6 <*ipv6-address*>	"Specifying an IPv6 SNMP trap receiver" on page 1073
web access-group ipv6 <ipv6 ACL name>	"Restricting web management access to an IPv6 host by specifying an IPv6 ACL" on page 1074
web client ipv6 <*ipv6-address*>	"Restricting web management access to an IPv6 host" on page 1074

IPv6 multicast

Commands	See ...
ipv6 router pim	"Enabling MLDv2" on page 1259
ipv6 pim-sparse	"Enabling MLDv2" on page 1259
ipv6 mld port-version <*version-number*>	"Enabling MLDv2" on page 1259
ssm-enable	"Enabling source specific multicast" on page 1259
ipv6 mld query-interval <*seconds*>	"Setting the query interval" on page 1260
ipv6 mld max-response-time <*seconds*>	"Setting the maximum response time" on page 1260
ipv6 mld llqc <*seconds*>	"Setting the last listener query count" on page 1260
ipv6 mld llqi <*seconds*>	"Setting the last listener query interval" on page 1260
ipv6 mld robustness<*seconds*>	"Setting the robustness" on page 1261
ipv6 mld version <*version-number*>	"Setting the version" on page 1261 "Setting the interface MLD version" on page 1262
mld port-ver <*version-number*>	"Specifying a port version" on page 1261
ipv6 mld static-group <*multicast-group-address*> [ethernet <*port-number*> [ethernet <*port-number*> | to <*port-number*>]*]	"Specifying a static group" on page 1261
show ipv6 mld group	"Displaying MLD group information" on page 1262
show ipv6 mld interface [<*port-number*>]	"Displaying MLD definitions for an interface" on page 1263
show ipv6 mld traffic	"Displaying MLD traffic" on page 1264
clear ipv6 mld traffic ethernet <*slot-number*>/<*port-number*> | ve <*ve-number*>	"Clearing IPv6 MLD traffic" on page 1264

IPv6 RIPng

Commands	See ...
clear ipv6 rip routes	"Clearing RIPng routes from IPv6 route table" on page 1115
distribute-list prefix-list <*name*> in | out <*interface*> <*port*>	"Controlling distribution of routes through RIPng" on page 1113
ipv6 rip default-information only | originate	"Configuring default route learning and advertising" on page 1112
ipv6 rip enable	"Configuring RIPng" on page 1109
ipv6 rip metric-offset [out] <1 – 16>	"Changing the metric of routes learned and advertised on an interface" on page 1113
ipv6 rip summary-address <*ipv6-prefix*>/<*prefix-length*>	"Advertising IPv6 address summaries" on page 1112
ipv6 router rip	"Enabling RIPng" on page 1110

Commands	See ...
poison-local-routes	"Configuring poison reverse parameters" on page 1114
poison-reverse	"Configuring poison reverse parameters" on page 1114
redistribute bgp | connected | isis | ospf | static [metric <*number*>]	"Redistributing routes into RIPng" on page 1113
show ipv6 rip	"Displaying RIPng configuration" on page 1115
show ipv6 rip route [<*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*>]	"Displaying RIPng routing table" on page 1116
timers <*update-timer*> <*timeout-timer*> <*hold-down-timer*> <*garbage-collection-timer*>	"Configuring RIPng timers" on page 1110

IPv6 OSPFv3

Commands	See ...
area <*number*> | <*ipv4-address*>	"Assigning OSPFv3 areas" on page 1201
area <*number*> | <*ipv4-address*> stub <*metric*> [no-summary]	"Assigning a totally stubby area" on page 1202
area <*number*> | <*ipv4-address*> virtual-link <*router-id*>	"Configuring virtual links" on page 1203
area <*number*> | <*ipv4-address*> virtual-link <*router-id*> [dead-interval <*seconds*> | hello-interval <*seconds*> | retransmit-interval <*seconds*> | transmit-delay <*seconds*>]	"Modifying virtual link parameters" on page 1204
auto-cost reference-bandwidth <*number*>	"Changing the reference bandwidth for the cost on OSPFv3 interfaces" on page 1205
auto-cost reference-bandwidth <*number*>	"Modifying exit overflow interval" on page 1216
default-information-originate [always] [metric <*value*>] [metric-type <*type*>]	"Configuring default route origination" on page 1213
default-metric <*number*>	"Modifying default metric for routes redistributed into OSPF version 3" on page 1208
distance external | inter-area | intra-area <*distance*>	"Configuring administrative distance based on route type" on page 1215
distribute-list prefix-list <*name*> in [<*interface*>]	"Configuring an OSPFv3 distribution list using an IPv6 prefix list as input" on page 1211
distribute-list route-map <*name*> in	"Configuring an OSPFv3 distribution list using an IPv6 prefix list as input" on page 1211
external-lsdb-limit <*entries*>	"Modifying external link state database limit" on page 1216
ipv6 ospf area <*number*> | <*ipv4-address*>	"Assigning interfaces to an area" on page 1203
pv6 prefix-list <*name*> [seq <*seq-value*>] [description <*string*>] deny | permit <*ipv6-addr*>/<*mask-bits*> [ge <*ge-value*>] [le <*le-value*>]	"Configuring an OSPFv3 distribution list using a route map as input" on page 1212

Commands	See ...
pv6 router ospf	"Enabling OSPFv3" on page 1201
log-status-change	"Disabling or reenabling event logging" on page 1218
metric-type type1 | type2	"Modifying metric type for routes redistributed into OSPF version 3" on page 1209
redistribute bgp | connected | isis [level-1 | level-1-2 | level-2] | rip | static [metric <*number*> | metric-type <*type*>]	"Redistributing routes into OSPFv3" on page 1206
redistribute bgp | connected | isis | rip | static [route-map <*map-name*>]	"Redistributing routes into OSPFv3" on page 1206
show ipv6 ospf area [<*area-id*>]	"Displaying OSPFv3 area information" on page 1218
show ipv6 ospf database [advrtr <*ipv4-address*> | as-external | extensive | inter-prefix | inter-router | intra-prefix | link | link-id <*number*> | network | router [scope <*area-id*> | as | link]]	"Displaying OSPFv3 database Information" on page 1219
show ipv6 ospf interface [ethernet <*port*> | loopback <*number*> | tunnel <*number*> | ve <*number*>]	"Displaying OSPFv3 interface information" on page 1224
show ipv6 ospf memory	"Displaying OSPFv3 memory usage" on page 1227
show ipv6 ospf neighbor [router-id <*ipv4-address*>]	"Displaying OSPFv3 neighbor information" on page 1228
show ipv6 ospf redistribute route [<*ipv6-prefix*>]	"Displaying routes redistributed into OSPFv3" on page 1230
show ipv6 ospf routes [<*ipv6-prefix*>]	"Displaying OSPFv3 route information" on page 1231
show ipv6 ospf spf node area [<*area-id*>]	"Displaying OSPFv3 SPF information" on page 1233
show ipv6 ospf spf table area <*area-id*>	"Displaying OSPFv3 SPF information" on page 1233
show ipv6 ospf spf tree area <*area-id*>	"Displaying OSPFv3 SPF information" on page 1233
show ipv6 ospf virtual-link	"Displaying IPv6 OSPF virtual link information" on page 1236
show ipv6 ospf virtual-neighbor	"Displaying IPv6 OSPF virtual link information" on page 1236
summary-address <*ipv6-prefix*>/<*prefix-length*>	"Configuring external route summarization" on page 1209
timers lsa-group-pacing <*seconds*>	"Configuring the OSPFv3 LSA pacing interval" on page 1216
timers spf <*delay*> <*hold-time*>	"Modifying shortest path first timers" on page 1214
virtual-link-if-address interface ethernet <*port*> | loopback <*number*> | tunnel <*number*> | ve <*number*>	"Assigning a virtual link source address" on page 1204

IS-IS

Commands	See ...
address-family ipv4 unicast	"Address family configuration level" on page 884
area-password <*string*>	"Configuring an area password" on page 888
clear isis all | counts | neighbor | route | traffic	"Clearing IS-IS information" on page 914
csnp-interval <*secs*>	"Changing the sequence numbers PDU interval" on page 889
default-information-originate [route-map <*name*>]	"Enabling advertisement of a default route" on page 893
default-metric <*number*>	"Changing the default redistribution metric" on page 896
disable-partial-spf-opt	"Disabling partial SPF calculations" on page 892
distance <*number*>	"Changing the administrative distance for IPv4 IS-IS" on page 894
domain-password <*string*>	"Configuring a domain password" on page 888
exit-address-family	"Address family configuration level" on page 884
hello padding [point-to-point]	"Globally disabling or re-enabling hello padding" on page 891 "Disabling and enabling hello padding on an interface" on page 901
hostname	"Disabling or re-enabling display of hostname" on page 889
ip router isis	"Disabling and enabling IS-IS on an interface" on page 899
ipv4 router isis	"Interface level" on page 885
isis circuit-type level-1 | level-1-2 | level-2	"Changing the IS-IS level on an interface" on page 900
isis hello-interval <*num*> [level-1-only | level-2-only]	"Changing the hello interval" on page 901
isis hello-multiplier <*num*> [level-1-only | level-2-only]	"Changing the hello multiplier" on page 901
isis metric <*num*>	"Changing the metric added to advertised routes" on page 902
isis passive	"Disabling or re-enabling formation of adjacencies" on page 899
isis password <*string*>	"Limiting access to adjacencies with a neighbor" on page 900
isis priority <*num*> [level-1-only | level-2-only]	"Setting the priority for designated IS election" on page 900
is-type level-1-only | level-1-2 | level-2-only	"Changing the IS-IS Level globally" on page 888
log-adjacency-changes	"Logging adjacency changes" on page 892
lsp interval <*seconds*>	"Changing the LSP interval and retransmit interval" on page 891
lsp-gen-interval <*secs*>	"Changing the LSP generation interval" on page 890

Commands	See ...
lsp-refresh-interval <*secs*>	"Changing the LSP refresh interval" on page 890
maximum-paths <*number*>	"Changing the maximum number of load sharing paths" on page 893
max-lsp-lifetime <*secs*>	"Changing the maximum LSP lifetime" on page 890
metric-style wide [level-1-only | level-2-only]	"Changing the metric style" on page 893
net <*area-id*>.<*system-id*>.<*sel*>	"Enabling IS-IS globally" on page 885 "Disabling and enabling IS-IS on an interface" on page 899
redistribute bgp [level-1 | level-1-2 | level-2 | metric <*number*> | metric-type external | internal | route-map <*name*>]	"Redistributing BGP4+ routes into IPv4 IS-IS" on page 898
redistribute connected [level-1 | level-1-2 | level-2 | metric <*number*> | metric-type external | internal | route-map <*name*>]	"Redistributing directly connected routes into IPv4 IS-IS" on page 897
redistribute isis level-1 into level-2 | level-2 into level-1 [prefix-list <*name*>]	"Redistributing IPv4 IS-IS routes within IPv4 IS-IS" on page 898
redistribute ospf [level-1 | level-1-2 | level-2 | match external1 | external2 | internal | metric <*number*> | metric-type external | internal | route-map <*name*>]	"Redistributing OSPF routes into IPv4 IS-IS" on page 898
redistribute rip [level-1 | level-1-2 | level-2 | metric <*number*> | metric-type external | internal | route-map <*name*>]	"Redistributing RIP routes into IPv4 IS-IS" on page 897
redistribute static [level-1 | level-1-2 | level-2 | metric <*number*> | metric-type external | internal | route-map <*name*>]	"Redistributing static IPv4 routes into IPv4 IS-IS" on page 896
retransmit-interval	"Changing the LSP interval and retransmit interval" on page 891
route-map <*map-name*> permit | deny <*sequence-number*>	"Enabling advertisement of a default route" on page 893
router isis	"Global configuration level" on page 884 "Disabling and enabling IS-IS on an interface" on page 899
set level level-1 | level-1-2 | level-2	"Enabling advertisement of a default route" on page 893
set-overload-bit [on-startup <*secs*>]	"Setting the overload bit" on page 887
show isis config	"Displaying the IS-IS configuration in the running-config" on page 903
show isis counts	"Displaying error statistics" on page 913
show isis database [<*lsp-id*> | detail | l1 | l2 | level1 | level2]	"Displaying summary information" on page 910
show isis database detail [l1 | l2 | level1 | level2]	"Displaying detailed information" on page 911
show isis hostname	"Displaying the name mappings" on page 903
show isis interface	"Displaying interface information" on page 906
show isis neighbor	"Displaying neighbor information" on page 904

Commands	See ...
show isis routes	"Displaying route information" on page 908
show isis traffic	"Displaying traffic statistics" on page 912
show logging	"Displaying IS-IS Syslog messages" on page 905
spf-interval <*secs*>	"Changing the SPF timer" on page 891
summary-address <*ip-addr*> <*ip-mask*> [level-1-only | level-1-2 | level-2-only]	"Configuring summary addresses" on page 895

Metro ring

Commands	See ...
diagnostics	"Enabling MRP diagnostics" on page 404
enable	"Configuring MRP with shared interfaces" on page 403
hello-time <*ms*>	"Changing the hello and preforwarding times" on page 397
master	"Adding an MRP ring to a VLAN" on page 396
metro-ring <*ring-id*>	"Adding an MRP ring to a VLAN" on page 396
name <*string*>	"Adding an MRP ring to a VLAN" on page 396
preforwarding-time <*ms*>	"Changing the hello and preforwarding times" on page 397
ring-interface ethernet <*primary-if*> ethernet <*secondary-if*>	"Adding an MRP ring to a VLAN" on page 396
show metro [<*ring-id*>]	"Displaying ring information" on page 406
show metro <*ring-id*> diag	"Enabling MRP diagnostics" on page 404
show topology-group [<*group-id*>]	"Displaying topology group information" on page 405

MSTP

Commands	See ...
mstp name <*name*>	"Setting the MSTP name" on page 1039
mstp revision <*revision-number*>	"Setting the MSTP revision number" on page 1039
mstp instance <*instance-number*> [vlan <*vlan-id*> | vlan-group <*group-id*>	"Configuring an MSTP instance" on page 1040
mstp instance <*instance-number*> ethernet <*slot/port*> priority <*port-priority*> path-cost <*cost*>	"Configuring port priority and port path cost" on page 1040
mstp instance <*instance-number*> priority <*priority-value*>	"Configuring bridge priority for an MSTP instance" on page 1040

Commands	See ...
mstp force-version <*mode-number*> forward-delay <*value*> hello-time <*value*> max-age <*value*> max-hops <*value*>	"Setting the MSTP global parameters" on page 1041
mstp admin-edge-port ethernet <*slot/port*>	"Setting ports to be operational edge ports" on page 1041
mstp admin-pt2pt-mac ethernet <*slot/port*>	"Setting point-to-point link" on page 1042
mstp disable <*slot/port*>	"Disabling MSTP on a port" on page 1042
mstp force-migration-check ethernet <*slot/port*>	"Forcing ports to transmit an MSTP BPDU" on page 1042
start	"Enabling MSTP on a switch" on page 1042
show mstp <*instance-number*>	"Displaying MSTP statistics" on page 1045
show mstp [<*mstp-id*> | configuration | detail] [| begin <*string*> | exclude <*string*> | include <*string*>] Enter an MSTP ID for <*mstp-id*>	"Displaying MSTP statistics" on page 1045 "Displaying MSTP information for a specified instance" on page 1047 "Displaying MSTP information for CIST instance 0" on page 1047

Multicast (IP)

Commands	See ...
bsr-candidate ethernet <*portnum*> | loopback <*num*> | ve <*num*> <*hash-mask-length*> [<*priority*>]	"Configuring BSRs" on page 600
clear pim rp-map	"Configuring RPs" on page 601
default-gateway <*ip-addr*>	"Modifying default route" on page 649
disable-dvmrp	"Globally enabling or disabling DVMRP without deleting multicast configuration" on page 647
graft-retransmit-time <*5-3600*>	"Modifying graft retransmit time" on page 649
graft-retransmit-timer <*10-3600*>	"Modifying graft retransmit timer" on page 595
hello-timer <*10-3600*>	"Modifying hello timer" on page 594
inactivity-timer <*10-3600*>	"Modifying inactivity timer" on page 595
ip dvmrp	"Enabling DVMRP on an interface" on page 647
ip dvmrp metric <*1-31*> | ttl-threshold <*1-64*> | [advertise-local on | off]	"Modifying the metric" on page 650 "Modifying the TTL" on page 650 "Enabling advertising" on page 650
ip igmp group-membership-time <*1-7200*>	"Modifying IGMP (V1 and V2) membership time" on page 576
ip igmp max-response-time <*num*>	"Modifying IGMP (V1 and V2) maximum response time" on page 576
ip igmp query-interval <*1-3600*>	"Modifying IGMP (V1 and V2) query interval period" on page 576

Commands	See ...
ip igmp static-group <*ip-addr*> [ethernet <*portnum*>]	"Adding an interface to a multicast group" on page 577
ip multicast-routing	"Changing IGMP V1 and V2 parameters" on page 575
ip pim [version 1 | 2]	"Enabling a PIM version" on page 593
ip pim border	p. 23-16
ip pim ttl <*1-64*>	"Modifying the TTL" on page 596
ip pim-sparse	"Configuring PIM interface parameters" on page 599
message-interval <*num*>	"Changing the PIM join and prune message interval" on page 609
ip mroute <*ip-addr*> <*ip-mask*> [<next-hop-ip-addr> | ethernet <*slot/port*> | ve <*num*> | null0] [<*cost*>] [distance < *num*>] }	"Configuring a static multicast route" on page 651
nbr-timeout	"Modifying neighbor timeout" on page 593 (PIM) "Modifying neighbor timeout" on page 648 (DVMRP)
probe-interval <*5-30*>	"Modifying probe interval" on page 649
prune-age <*20-3600*>	"Modifying prune age" on page 648
prune-timer <*10-3600*>	"Modifying prune timer" on page 594
prune-wait <*time*>	"Modifying the prune wait timer" on page 594
report-interval <*10-2000*>	"Enabling DVMRP on an interface" on page 647
route-discard-timeout <*40-8000*>	"Modifying route discard time" on page 648
route-expire-timeout <*20-4000*>	"Modifying route expires time" on page 648
router dvmrp	"Globally enabling and disabling DVMRP" on page 647
router pim	"Globally enabling and disabling PIM" on page 592 "Configuring global PIM Sparse parameters" on page 599
rp-address <*ip-addr*>	"Statically specifying the RP" on page 602
rp-candidate add l delete <*group-addr*> <*mask-bits*>	"Configuring RPs" on page 601
rp-candidate ethernet <*slot*>/<*portnum*> | loopback <*num*> | ve <*num*>	"Configuring RPs" on page 601
show ip pim | dvmrp rpf <*IP address*>	"Displaying information about an upstream neighbor device" on page 651 "Displaying information about an upstream neighbor device" on page 617
show ip pim bsr	"Displaying BSR information" on page 613
show ip pim dense	"Viewing the prune wait time" on page 595
show ip pim group	"Displaying a list of multicast groups" on page 612
show ip pim mcache	"Displaying the PIM multicast cache" on page 618
show ip pim nbr	"Displaying multicast neighbor information" on page 616

Commands	See ...
show ip pim rp-candidate	"Displaying candidate RP information" on page 614
show ip pim rp-hash <*group-addr*>	"Displaying RP information for a PIM Sparse group" on page 615
show ip pim rp-map	"Displaying RP-to-group mappings" on page 615
show ip pim rp-set	"Displaying the RP set list" on page 616
show ip pim sparse	"Displaying basic PIM Sparse configuration information" on page 611
show ip pim traffic	"Displaying PIM traffic statistics" on page 620
spt-threshold infinity | <*num*>	"Changing the Shortest Path Tree (SPT) threshold" on page 609
system-max dvmrp-mcache <*num*>	"Defining the maximum number of DVMRP cache entries" on page 573
system-max pim-mcache <*num*>	"Defining the maximum number of PIM cache entries" on page 573
trigger-interval <*5-30*>	"Modifying trigger interval" on page 649

Multicast (L2)

Commands	See ...
clear ip multicast all | group <*group-id*>	"Clearing IGMP group flows" on page 1061
clear ip multicast statistics	"Clearing IP multicast statistics" on page 1061
ip multicast [active | passive]	"Enabling IP multicast traffic reduction" on page 1050 "Changing the IGMP mode" on page 1051 "Enabling PIM SM traffic snooping" on page 1058
ip multicast age-interval <*interval*>	"Modifying the age interval" on page 1052
ip multicast filter	"Filtering multicast groups" on page 1052
ip multicast query-interval <*interval*>	"Modifying the query interval" on page 1052
ip pimsm-snooping	"Enabling PIM SM traffic snooping" on page 1058
show ip multicast igmp-snooping	"Displaying multicast information" on page 1060
show ip multicast pimsm-snooping	"Displaying multicast information" on page 1060
show ip multicast statistics	"Displaying IP multicast statistics" on page 1061
show ip multicast	"Enabling IP multicast traffic reduction" on page 1050

OSPF version 4

Commands	See ...
area <*num*> | <*ip-addr*> nssa <*cost*> | default-information-originate	"Configuring an NSSA" on page 686
area <*num*> | <*ip-addr*> range <*ip-addr*> <*ip-mask*> [advertise | not-advertise]	"Configuring an address range for the NSSA" on page 687 "Assigning an area range (optional)" on page 687
area <*num*> | <*ip-addr*> stub <*cost*> [no-summary]	"Assign a totally stubby area" on page 684
area <*num*> | <*ip-addr*> virtual-link <*ip-addr*> [authentication-key [0 | 1] <*string*>] [dead-interval <*num*>] [hello-interval <*num*>] [md5-authentication key-activation-wait-time <*num*> | key-id <*num*> [0 | 1] key <*string*>] [retransmit-interval <*num*>] [transmit-delay <*num*>]	"Modify virtual link parameters" on page 694
area <*num*> | <*ip-addr*> virtual-link <*router-id*> [authentication-key | dead-interval | hello-interval | retransmit-interval | transmit-delay <*value*> | [md5-authentication key-activation-wait-time <*num*> | key-id <*num*> [0 | 1] key <*string*>]	"Assign virtual links" on page 692
area <*num*> | <*ip-addr*>	"Assign OSPF areas" on page 683
auto-cost reference-bandwidth <*num*>	"Changing the reference bandwidth" on page 700
database-overflow-interval <*value*>	"Modify exit overflow interval" on page 716
default-information-originate [always] [metric <*value*>] [metric-type <*type*>]	"Configure default route origination" on page 706
default-metric <*value*>	"Modify default metric for redistribution" on page 702
distance external | inter-area | intra-area <*distance*>	"Modify administrative distance" on page 709
ip ospf area <*ip-addr*>	"Assigning interfaces to an area" on page 688
ip ospf auth-change-wait-time <*secs*>	"Change the timer for OSPF authentication changes" on page 691
ip ospf database-filter all out	"Block flooding of outbound LSAs on specific OSPF interfaces" on page 691
log all | adjacency | bad_packet [checksum] | database | memory | retransmit	"Specify types of OSPF Syslog messages to log" on page 716
metric-type type1 | type2	"Modify redistribution metric type" on page 708
redistribution bgp | connected | rip | [isis level-1| level-1-2| level-2] | static [route-map <*map-name*>]	"Enable route redistribution" on page 702 "Define redistribution filters" on page 700
rfc1583-compatibility	"Modify OSPF standard compliance setting" on page 716
router ospf	"Enable OSPF on the router" on page 683
show ip ospf area [<*area-id*>] | [<*num*>]	"Displaying OSPF area information" on page 720
show ip ospf border-routers [<*ip-addr*>]	"Displaying OSPF ABR and ASBR information" on page 729

Commands	See ...
show ip ospf config	"Configure external route summarization" on page 705 "Displaying general OSPF configuration information" on page 718
show ip ospf database external-link-state [advertise <*num*>] | [extensive] | [link-state-id <*ip-addr*>] | [router-id <*ip-addr*>] | [sequence-number <num(Hex)>] | [status <*num*>]	"Displaying OSPF external link state Information" on page 727
show ip ospf database link-state [advertise <*num*>] | [asbr] | [extensive] | [link-state-id <*ip-addr*>] | [network] | [nssa] | [router] | [router-id <*ip-addr*>] | [sequence-number <num(Hex)>] | [summary]	"Displaying OSPF database link state information" on page 728
show ip ospf interface [<*ip-addr*>]	"Displaying OSPF interface information" on page 723
show ip ospf neighbor [router-id <*ip-addr*>] | [<*num*>]	"Displaying OSPF neighbor information" on page 721
show ip ospf redistribute route [<*ip-addr*> <*ip-mask*>]	"Displaying the routes that have been redistributed into OSPF" on page 726
show ip ospf routes [<*ip-addr*>]	"Displaying OSPF route information" on page 725
show ip ospf trap	"Displaying OSPF trap status" on page 730
show ip ospf virtual link [<*num*>]	"Displaying OSPF virtual link information" on page 732
show ip ospf virtual neighbor [<*num*>]	"Displaying OSPF virtual neighbor and link information" on page 730
show tasks	"Displaying CPU utilization and other OSPF tasks" on page 719
snmp-server trap ospf <*ospf-trap*>	"Modifing OSPF traps generated" on page 714
summary-address <*ip-addr*> <*ip-mask*>	"Configure external route summarization" on page 705
timers lsa-group-pacing <*secs*>	"Changing the LSA pacing interval" on page 710
timers spf <*delay*> <*hold-time*>	"Modify SPF timers" on page 708

Port parameters

Commands	See ...
config-trunk-ind	"Monitoring an individual trunk port" on page 142
disable	"Disabling or re-enabling a port" on page 135
enable	"Disabling or re-enabling a port" on page 135
flow-control	"Disabling or re-enabling flow control" on page 136
gig-default neg-full-auto | auto-gig | neg-off	"Changing the default Gigabit negotiation mode" on page 136 "Changing the default Gigabit negotiation mode" on page 136

Commands	See ...
ip address <*ip-addr*> <*ip-mask*> | <*ip-addr*>/<*mask-bits*>	"Assigning an IP address to a port" on page 134
lock-address ethernet <*portnum*> [addr-count <*num*>]	"Locking a port to restrict addresses" on page 137
monitor ethe-port-monitored <*portnum*> | named-port-monitored <*portname*> ethernet | <*portnum*> in | out | both	"Monitoring an individual trunk port" on page 142
phy-mode wan	"Enabling WAN PHY mode support" on page 144
port-name <*text*>	"Assigning a port name" on page 133
qd-flow sink <*sinking-threshold*> sunk <*sunk-threshold*> slot <*slot*>	"Specifying threshold values for flow control" on page 137
set mirror-interface <*slot number*>/<*port number*>	"Configuring mirror ports for PBR traffic" on page 143
show monitor actual	"Displaying mirror and monitor port configuration" on page 144
show monitor config	"Displaying mirror and monitor port configuration" on page 144
speed-duplex <*value*>	"Speed/Duplex negotiation" on page 134

Port-based routing

Commands	See ...
access-list <*num*> deny | permit <*source-ip*> | <*hostname*> <*wildcard*>	"Configure the ACLs" on page 564
access-list <*num*> deny | permit <*source-ip*>/<*mask-bits*> | <*hostname*>	
access-list <*num*> deny | permit any	
access-list <*num*> deny | permit host <*source-ip*> | <*hostname*>	
ip policy route-map <*map-name*>	"Enabling PBR" on page 566
match ip address <ACL-num-or-name>	"Configure the route map" on page 566
route-map <*map-name*> permit | deny <*num*>	"Configure the route map" on page 566 "Basic example" on page 567
set interface null0	"Configure the route map" on page 566
set ip next hop <*ip-addr*>	"Configure the route map" on page 566 "Basic example" on page 567

Quality of Service (QoS)

Commands	See ...
dscp <*num*>	"Configuring DSCP classification by interface" on page 468
priority <*num*>	"Changing a port's priority" on page 469 "Changing a Layer 2 port-based VLAN's priority" on page 469
qos multicast best-effort rate <*rate*>	"Configuring multicast traffic engineering" on page 486
qos queue-type <*queue-number*> wred averaging-weight <*avg-weight*>	"Setting the averaging-weight (Wq) parameter" on page 478
qos queue-type <*queue-number*> wred drop-precedence <*policing-status*> drop-probability-max <*p-max*>	"Setting the maximum drop probability" on page 479
qos queue-type <*queue-number*> wred drop-precedence <*policing-status*> max-avg-queue-size <*max-size*>	"Setting the minimum and maximum average queue size" on page 479
qos queue-type <*queue-number*> wred drop-precedence <*policing-status*> min-avg-queue-size <*min-size*>	"Setting the minimum and maximum average queue size" on page 479
qos queue-type <*queue-number*> max-queue-size <*max-queue*>	"Configuring the maximum instantaneous queue size" on page 479
qos queue-type <*queue-number*> wred drop-precedence <*policing-status*> packet-size-max <*pkt-size*>	"Setting the maximum packet size" on page 480
qos queue-type <*queue-number*> wred enable	"Enabling WRED" on page 477
qos scheduler destination-weighted <*queue0-weight*> <*queue1-weight*> <*queue2-weight*> <*queue3-weight*>	"Configuring WFQ destination-based traffic scheduling" on page 484
qos scheduler enhanced-strict <*queue0-rate*> <*queue1-rate*> <*queue2-rate*> <*queue3-rate*>	"Configuring enhanced strict priority-based traffic scheduling" on page 483
qos scheduler max-rate <*queue0-rate*> <*queue1-rate*> <*queue2-rate*> <*queue3-rate*>	"Configuring maximum rate-based traffic scheduling" on page 485
qos scheduler min-rate <*queue0-rate*> <*queue1-rate*> <*queue2-rate*> <*queue3-rate*>	"Configuring minimum rate-based traffic scheduling" on page 485
qos scheduler source-weighted <*queue0-weight*> <*queue1-weight*> <*queue2-weight*> <*queue3-weight*>	"Configuring WFQ source-based traffic scheduling" on page 484
qos scheduler strict	"Configuring strict priority-based traffic scheduling" on page 483
qos-tos map cos-dscp <*dscp0*> <*dscp1*> <*dscp2*> <*dscp3*> <*dscp4*> <*dscp5*> <*dscp6*> <*dscp7*>	"Changing the CoS –> DSCP mappings" on page 471
qos-tos map cos-priority <*prio0*> <*prio1*><*prio2*><*prio3*><*prio4*><*prio5*><*prio6*> <*prio7*>	"Changing the DSCP –> internal forwarding priority mappings" on page 472
qos-tos map dscp-dscp <*old-dscp-value*> [<*old-dscp-value*>...] to <*new-dscp-value*>	"Changing the DSCP –> DSCP mappings" on page 472
qos-tos map dscp-priority <*dscp-value*> [<*dscp-value*> ...] to <*priority*>	"Changing the DSCP –> internal forwarding priority mappings" on page 472
qos-tos mark cos | dscp	"Enabling marking" on page 471

Commands	See ...
qos-tos trust cos | dscp	"Specifying trust level" on page 470
qos-tos	"Enabling ToS-based QoS" on page 470
show qos multicast [port-no]	"Displaying the multicast traffic engineering configuration" on page 487
show qos scheduler	"Displaying the scheduler configuration" on page 486
show qos wred	"Displaying the WRED configuration" on page 481
show qos-tos	"Displaying QoS configuration information" on page 474
static-mac-address <*mac-addr*> ethernet <*portnum*> [priority <*num*>] [host-type | router-type | fixed-host]	"Assigning static MAC address entries to priority queues" on page 470

Rate limiting

Commands	See ...
rate-limit in access-group <*number*> | named-access-group <*acl-name*> <*average-rate*> <*maximum-burst*>	"Configuring a port-and-ACL-based traffic policing policy" on page 501
rate-limit in ipv6-named-access-group <*name*> <*average-rate*> <*maximum-burst*>	"Configuring a port-and-priority-based rate limiting policy" on page 499
rate-limit in group <*group-number*> <*average-rate*> <*maximum-burst*>	"Configuring a VLAN-group-based rate limiting policy" on page 500
rate-limit in group <*group-number*> priority <*num*> <*average-rate*> <*maximum-burst*>	"Configuring a VLAN-group-based rate limiting policy" on page 500
rate-limit input priority <*num*> <*average-rate*> <*maximum-burst*>	"Configuring a port-and-priority-based rate limiting policy" on page 499
rate-limit input <*average-rate*> <*maximum-burst*>	"Configuring a port-based rate limiting policy" on page 498
rate-limit input vlan <*vlan-number*> <*average-rate*> <*maximum-burst*>	"Configuring a port-and-VLAN-based rate limiting policy" on page 499
rate-limit strict-acl	"Configuring a port-and-IPv6 ACL-based traffic reduction" on page 502
rl-vlan-group <*vlan-group-number*>	"Configuring a VLAN-group-based rate limiting policy" on page 500
show rate-limit [counters [interface <*slot/port*>]] [group [<*vlan-number*>]] [interface <*slot/port*>]	"Displaying traffic reduction" on page 504
vlan <*vlan-number*> [to <vlan-number]	"Configuring a VLAN-group-based rate limiting policy" on page 500

RIP

Commands	See ...
clear ip rip local	"Clearing the RIP routes from the routing table" on page 673
clear ip rip routes <*ip-addr*> / <*mask-bits*>	"Clearing the RIP routes from the routing table" on page 673
default-metric <*1-15*>	"Changing the default redistribution metric" on page 668
distance <*num*>	"Changing the administrative distance" on page 666
ip prefix-list <*name*> permit | deny <*source-ip-address*> | any <*source-mask*> | any	"Using prefix lists and route maps as route filters" on page 671
ip rip learn-default	"Configuring route learning and advertising parameters" on page 668
ip rip metric-offset <*num*> in | out	"Changing the cost of routes learned or advertised on a port" on page 666
ip rip poison-reverse	"Changing the route loop prevention method" on page 669
ip rip prefix-list <*name*> in | out	"Using prefix lists and route maps as route filters" on page 671
ip rip route-map <*name*> in | out	"Using prefix lists and route maps as route filters" on page 671
ip rip v1-only | v1-compatible-v2 | v2-only	"Enabling RIP" on page 666
learn-default	"Configuring route learning and advertising parameters" on page 668
neighbor <*filter-num*> permit | deny <*source-ip-address*> | any	"Configuring a RIP neighbor filter" on page 669
poison-local-routes	"Changing the route loop prevention method" on page 669
poison-reverse	"Changing the route loop prevention method" on page 669
prefix-list <*name*> in | out	"Using prefix lists and route maps as route filters" on page 671
redistribute connected | bgp | ospf | static [metric <*value*> | route-map <*name*>]	"Configuring redistribution filters" on page 667
router rip	"Enabling RIP" on page 666
show ip rip	"Displaying RIP filters" on page 672
timers <*seconds*>	"Setting RIP timers" on page 672

RMON

Commands	See ...
clear <*option*>	"Clearing statistics" on page 1020
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*>	"Alarm (RMON group 3)" on page 1023
rmon event <*event-entry*> description <*text-string*> log | trap | log-and-trap | owner <*rmon-station*>	"Event (RMON group 9)" on page 1024
rmon history <*entry-number*> interface ethernet<*slot/port*> | management <*num*> buckets <*number*> interval <*sampling-interval*> owner <*text-string*>	"History (RMON group 2)" on page 1023
show <*option*>	"Viewing configuration information" on page 1019
show rmon statistics [<*num*> | ethernet <*slot/port*> | management <*num*> | | begin <*expression*> | exclude <*expression*> | include <*expression*>]	"Statistics (RMON group 1)" on page 1020
show version	"Viewing system information" on page 1019

RSTP

Commands	See ...
rstp [admin-edge-port] | [admin-pt2pt-mac]	"Changing port parameters" on page 377
rstp ethernet <*portnum*> path-cost <*value*> | priority <*value*> | [admin-edge-port] | [admin-pt2pt-mac] | [force-migration-check]	"Changing port parameters" on page 377
rstp single	"Enabling or disabling RSTP on a single spanning tree" on page 376
rstp	"Enabling or disabling RSTP in a port-based VLAN" on page 375
show rstp [vlan <*vlan-id*>]	"Displaying RSTP information" on page 383
show rstp detail [vlan <*vlan-id*>]	"Displaying RSTP information" on page 383
spanning-tree 802-1w [forward-delay <*value*>] | [hello-time <*value*>] | [max-age <*time*>] | [force-version <*value*>] | [priority <*value*>]	"Changing RSTP bridge parameters" on page 376
spanning-tree	"Disabling or enabling RSTP on a port" on page 376

Security/management

802.1x port security

Commands	See ...
aaa authentication dot1x default <*method-list*>	"Configuring an authentication method list for 802.1x" on page 961
auth-fail-action restricted-vlan	"Specifying the authentication-failure action" on page 971
auth-fail-max-attempts <*attempts*>	"Specifying the number of authentication attempts the device makes before dropping packets" on page 971
auth-fail-vlanid <*vlan-id*>	"Specifying the authentication-failure action" on page 971
clear dot1x mac-session <*mac-address*>	"Clearing a dot1x-mac-session for a MAC address" on page 971
clear dot1x statistics all I <*portnum*>	"Clearing 802.1x statistics" on page 975
dot1x filter-strict-security	"Disabling and enabling strict security mode for dynamic filter assignment" on page 963
dot1x initialize <*portnum*>	"Initializing 802.1x on a port" on page 970
dot1x port-control [force-authorized | force-unauthorized | auto]	"Setting the port control" on page 967
dot1x re-authenticate <*portnum*>	"Re-authenticating a port manually" on page 968
dot1x-enable	"Enabling 802.1x port security" on page 966
enable all I<*portnum*> [to <*portnum*>]	"Enabling 802.1x port security" on page 966
global-filter-strict-security	"Disabling and enabling strict security mode for dynamic filter assignment" on page 963
maxreq <*value*>	"Specifying the number of EAP-request/identity frame retransmissions" on page 969
radius-server host <*ip-addr*> | <*server-name*> [auth-port <*number*> acct-port <*number*> [authentication-only | accounting-only | default [key 0 | 1 <*string*> [dot1x]]]]	"Setting RADIUS parameters" on page 961
re-authentication	"Configuring periodic re-authentication" on page 968
servertimeout <*seconds*>	"Specifying a timeout for retransmission of messages to the authentication server" on page 970
show dot1x	"Displaying 802.1x configuration information" on page 972
show dot1x config ethernet <*slot/port*>	"Displaying 802.1x configuration information" on page 972
show dot1x ip-acl [all | ethernet <*slot/port*> | | begin <*expression*> | exclude <*expression*> | include <*expression*>]	"Displaying IP ACLs applied to an 802.1x-enabled port" on page 977

Commands	See ...
show dot1x mac-address-filter [all | ethernet <*slot/port*> [begin <*expression*> | exclude <*expression*> | include <*expression*>]]	"Displaying MAC address filters applied to an 802.1x-enabled port" on page 976
show dot1x mac-session [brief [begin <*expression*> | exclude <*expression*> | include <*expression*>]]	"Displaying information about the dot1x-mac-sessions on each port" on page 978 "Displaying information about the ports in an 802.1x multiple client configuration" on page 979
show dot1x statistics [all | ethernet <*slot/port*>]	"Displaying 802.1x statistics" on page 974
supptimeout <*seconds*>	"Specifying a timeout for retransmission of EAP-request frames to the client" on page 970
timeout quiet-period <*seconds*>	"Setting the quiet period" on page 969
timeout re-authperiod <*seconds*>	"Configuring periodic re-authentication" on page 968
timeout tx-period <*seconds*>	"Setting the interval for retransmission of EAP-request/identity frames" on page 969

Access

Commands	See ...
all-client <*ip-addr*>	"Restricting all remote management access to a specific IP address" on page 67

Authentication method list

Commands	See ...
aaa authentication snmp-server | web-server | enable | login | dot1x default <*method1*> [<*method2*>] [<*method3*>] [<*method4*>] [<*method5*>] [<*method6*>] [<*method7*>]	"Examples of authentication-method lists" on page 111

Passwords

Commands	See ...
enable password-min-length <*number-of-characters*>	"Specifying a minimum password length" on page 75
enable port-config-password <*text*>	"Setting passwords for management privilege levels" on page 72
enable read-only-password <*text*>	"Setting passwords for management privilege levels" on page 72
enable super-user-password <*text*>	"Setting passwords for management privilege levels" on page 72
enable telnet password <*string*>	"Setting a Telnet password" on page 71
service password-encryption	"Disabling password encryption" on page 74

Privilege level

Commands	See ...
privilege <*cli-level*> level <*privilege-level*> <*command-string*>	"Augmenting management privilege levels" on page 73
show web	"Displaying TACACS/TACACS+ statistics and configuration information" on page 95 "Displaying RADIUS configuration information" on page 108

RADIUS

Commands	See ...
aaa accounting commands <*privilege-level*> default start-stop radius | none	"Configuring RADIUS accounting for CLI commands" on page 107
aaa accounting exec default start-stop radius | none	"Configuring RADIUS accounting for Telnet/SSH (Shell) access" on page 106
aaa accounting system default start-stop radius | none	"Configuring RADIUS accounting for system events" on page 107
aaa authentication enable implicit-user	"Configuring Enable authentication to prompt for password only" on page 104
aaa authentication login privilege-mode	"Entering privileged EXEC mode after a Telnet or SSH login" on page 104
aaa authorization commands <*privilege-level*> default radius | none	"Configuring command authorization" on page 105
aaa authorization exec default radius | none	"Configuring Exec authorization" on page 105
enable aaa console	"Command authorization and accounting for console commands" on page 106
radius-server host <*ip-addr*> | <*server-name*> [auth-port <*number*> acct-port <*number*> [authentication-only | authorization-only | accounting-only | default] [key <*string*>]]	"Identifying the RADIUS server to the BigIron RX" on page 101 "Specifying different servers for individual AAA functions" on page 102
radius-server key [0 | 1] <*string*>	"Setting the RADIUS key" on page 102
radius-server retransmit <*number*>	"Setting the retransmission limit" on page 87
radius-server timeout <*number*>	"Setting the timeout parameter" on page 103
show aaa	"Displaying RADIUS configuration information" on page 108

SNMP access

Commands	See ...
snmp-client <*ip-addr*>	"Restricting SNMP access to a specific IP address" on page 67
snmp-server community <*string*> ro | rw <*standard-acl-name*> | <*standard-acl-id*>	"Using ACLs to restrict SNMP access" on page 65
snmp-server enable vlan <*vlan-id*>	"Restricting SNMP access to a specific VLAN" on page 69
snmp-server enable	"Disabling SNMP access" on page 70

SSH access

Commands	See ...
ip ssh client <*ip-addr*>	"Restricting SSH access to a specific IP address" on page 67
ssh access-group <*num*> | <*name*>	"Using ACLs to restrict SNMP access" on page 65

SSL

Commands	See ...
crypto-ssl certificate generate	"Generating an SSL certificate" on page 79
crypto-ssl certificate zeroize	"Deleting the SSL certificate" on page 79
ip ssl certificate-data-file tftp <*ip-addr*> <*certificate-filename*>	"Importing digital certificates and RSA private key files" on page 79
ip ssl port <*port-number*>	"Specifying a port for SSL communication" on page 78
ip ssl private-key-file tftp <*ip-addr*> <*key-filename*>	"Importing digital certificates and RSA private key files" on page 79
web-management https	"Enabling the SSL server on the device" on page 78

TACACS/TACACS+

Commands	See ...
aaa accounting commands <*privilege-level*> default start-stop tacacs+ | none	"Configuring TACACS+ accounting for CLI commands" on page 93
aaa accounting exec default start-stop tacacs+ | none	"Configuring TACACS+ accounting for Telnet/SSH (Shell) access" on page 93
aaa accounting system default start-stop tacacs+ | none	"Configuring TACACS+ accounting for system events" on page 93
aaa authentication enable implicit-user	"Configuring Enable authentication to prompt for password only" on page 89

Commands	See ...
aaa authentication login privilege-mode	"Entering privileged EXEC mode after a Telnet or SSH login" on page 89
aaa authorization commands <*privilege-level*> default tacacs+ | none	"Configuring command authorization" on page 91
aaa authorization exec default tacacs+ | none	"Configuring Exec authorization" on page 89
enable aaa console	"AAA support for console commands" on page 92
show aaa	"Displaying TACACS/TACACS+ statistics and configuration information" on page 95
tacacs-server dead-time <*number*>	"Setting the dead time parameter" on page 87
tacacs-server host <*ip-addr*> | <*server-name*> [auth-port <*number*> [authentication-only | authorization-only | accounting-only | default] [key <*string*>]]	"Specifying different servers for individual AAA functions" on page 86
tacacs-server key [0 | 1] <*string*>	"Setting the TACACS+ key" on page 87
tacacs-server retransmit <*number*>	"Setting the retransmission limit" on page 87
tacacs-server timeout <*number*>	"Setting the timeout parameter" on page 88

Telnet access

Commands	See ...
telnet access-group <*num*> | <*name*>	"Using an ACL to restrict Telnet access" on page 64
telnet client <*ip-addr*>	"Restricting Telnet access to a specific IP address" on page 67
telnet login-retries <*number*>	"Specifying the maximum number of login attempts for Telnet access" on page 68
telnet server enable vlan <*vlan-id*>	"Restricting Telnet access to a specific VLAN" on page 68
telnet server suppress-reject-message	"Suppressing Telnet connection rejection messages" on page 71
telnet-server	"Disabling Telnet access" on page 70

TFTP access

Commands	See ...
tftp client enable vlan <*vlan-id*>	"Restricting TFTP access to a specific VLAN" on page 69

User account

Commands	See ...
username <*user-string*> privilege <*privilege-level*> password | nopassword <*password-string*>	"Configuring a local user account" on page 76
show users	"Configuring a local user account" on page 76

Web management access

Commands	See ...
web access-group <*num*> | <*name*>	"Using an ACL to restrict Web management access" on page 65
web client <*ip-addr*>	"Restricting Web Management access to a specific IP address" on page 67
web-management	"Disabling Web management access" on page 70
web-management enable vlan <*vlan-id*>	"Restricting Web Management access to a specific IP address" on page 67
web-management hp-top-tools	"Disabling Web management access by HP ProCurve Manager" on page 70
web-management http | https	"Enabling the SSL server on the device" on page 78

DoS protection

Commands	See ...
clear statistics dos-attack	"Clear DoS attack statistics" on page 988
dos-attack-prevent <*num*> burst-normal <*bps*> burst-max <*num-of-packets*> lockup <*seconds*> [log]	"Avoiding being a victim in a Smurf attack" on page 984
ip directed-broadcast	"Avoiding being an intermediary in a Smurf attack" on page 984
ip tcp tcp-security	"Disabling the TCP security enhancement" on page 987
show statistics dos-attack [| begin<*expression*> | exclude <*expression*> | include <*expression*>]	"Displaying statistics due DoS attacks" on page 988

MAC authentication

Commands	See ...
clear auth-mac-table <*portnum*>	"Clearing authenticated MAC addresses" on page 934
clear auth-mac-table	"Clearing authenticated MAC addresses" on page 934
mac-authentication apply-mac-auth-filter <*filter-id*>	"Defining MAC address filters" on page 932

Commands	See ...
mac-authentication auth-fail-action block-traffic	"Specifying the authentication-failure action" on page 931
mac-authentication auth-fail-action restrict-vlan [<*vlan-id*>]	"Specifying the authentication-failure action" on page 931
mac-authentication auth-fail-vlan-id <*vlan-id*>	"Specifying the authentication-failure action" on page 931
mac-authentication auth-passwd-format xxxx.xxxx.xxxx | xx-xx-xx-xx-xx-xx | xxxxxxxxxxx	"Specifying the format of the MAC addresses sent to the RADIUS server" on page 931
mac-authentication clear-mac-session <*mac-address*>	"Clearing authenticated MAC addresses" on page 934
mac-authentication disable-aging [denied-mac-only | permitted-mac-only]	"Disabling aging for authenticated MAC addresses" on page 935
mac-authentication enable [<*portnum*> | all]	"Enabling multi-device port authentication" on page 930
mac-authentication enable-dynamic-vlan	"Configuring dynamic VLAN assignment" on page 932
mac-authentication mac-filter <*filter*>	"Defining MAC address filters" on page 932
mac-authentication max-age <*seconds*>	"Specifying the aging time for blocked MAC addresses" on page 936
mac-authentication move-back-to-old-vlan disable | port-configured-vlan | port-restrict-vlan | system-default-vlan	"Specifying to which VLAN a port is moved after its RADIUS-specified VLAN assignment expires" on page 933
mac-authentication no-override-restrict-vlan	"Configuring dynamic VLAN assignment" on page 932
mac-authentication save-dynamicvlan-to-config	"Saving dynamic VLAN assignments to the running configuration file" on page 934
show auth-mac-address <*mac-address*> | <*ip-address*> | <*portnum*>	"Displaying multi-device port authentication information for a specific MAC address or port" on page 940
show auth-mac-address configuration	"Displaying multi-device port authentication configuration information" on page 937
show auth-mac-address detail	"Displaying multi-device port authentication configuration information" on page 937
show auth-mac-address	"Displaying authenticated MAC address information" on page 936
show auth-mac-addresses authorized-mac	"Displaying the authenticated MAC addresses" on page 941
show auth-mac-addresses unauthorized-mac	"Displaying the non-authenticated MAC addresses" on page 941

MAC port security

Commands	See ...
age <*minutes*>	"Setting the port security age timer" on page 945
autosave <*minutes*>	"Autosaving secure MAC addresses to the startup-config file" on page 945
enable	"Enabling the MAC port security feature" on page 944
	"Re-enabling a port" on page 947
maximum <*number-of-addresses*>	"Setting the maximum number of secure MAC addresses for an interface" on page 944
port security	"Enabling the MAC port security feature" on page 944
restrict-max-deny <*number*>	"Violation restrict" on page 946
secure <*mac-address*>	"Specifying secure MAC addresses" on page 945
show mac [all]	"Displaying a list of MAC addresses" on page 951
show port security <*module*> | <*portnum*>	"Displaying port security settings" on page 949
show port security mac	"Displaying the secure MAC addresses on the device" on page 950
show port security statistics <*portnum*> l <*module*>	"Displaying port security statistics" on page 950
shutdown-time <*minutes*>	"Port shutdown time" on page 946
violation restrict	"Violation restrict" on page 946
violation shutdown	"Violation shutdown" on page 946

Redundant management module

Commands	See ...
active-management <*mgt-module*>	"Changing the default active Chassis slot" on page 31
sync-standby	"Comparing and synchronizing files" on page 33
force-sync-standby	"Synchronizing files without comparison" on page 34
switchover	"Manually switching over to the standby management module" on page 34
reset	"Manually switching over to the standby management module" on page 34
boot system bootp | [flash primary | flash secondary] | slot <*number*> <*filename*> | tftp <*ip-address*> <*filename*>	"Rebooting the active and standby management modules" on page 34
reload	"Rebooting the active and standby management modules" on page 34
reboot-standby	"Rebooting the active and standby management modules" on page 34

Commands	See ...
show module	"Determining management module status" on page 35
show chassis	"Displaying temperature information" on page 36
format slot1 | slot2	"Formatting a flash card" on page 42
pwd	"Determining the current management focus" on page 42
cd <*directory-pathname*>	"Switching the management focus" on page 43
chdir <*directory-pathname*>	"Switching the management focus" on page 43
dir | ls [<*path-name*>]	"Displaying a directory of the files" on page 43
more [/<*directory*>/]<*file-name*>	"Displaying the contents of a file" on page 45
hd [/<*directory*>/]<*file-name*>	"Displaying the hexadecimal output of a file" on page 46
md | mkdir [slot1 | slot2] <*dir-name*>	"Creating a subdirectory" on page 46
rd | rmdir [slot1 | slot2] <*dir-name*>	"Removing a subdirectory" on page 48
rename | mv [/<*directory*>/]<*old-file-name*> [/<*directory*>/]<*new-file-name*>	"Renaming a file" on page 49
attrib [slot1 | slot2] ro | rw <*file-name*>	"Changing the read-write attribute of a file" on page 49
delete | rm [slot1 | slot2] [<*directory*>] [<*file-name*>]	"Deleting a file" on page 50
undelete	"Recovering ("undeleting") a file" on page 51
append [<*source-file-system*> <*dest-file-system*>] [/<*source-dir-path*>/]<*source-file-name*> [/<*dest-dir-path*>/]<*dest-file-name*>	"Appending a file to another file" on page 52
copy <*from-card*> <*to-card*> [/<*from-dir-path*>/]<*from-name*> [/<*to-dir-path*>/][<*to-name*>]	"Copying files using the copy command" on page 52
copy slot1 | slot2 flash [/<*from-dir-path*>/]<*from-name*> monitor | primary | secondary	"Copying files between a flash card and flash memory" on page 53
copy flash slot1 | slot2 <*source-name*> monitor | primary | secondary | startup-config [<*dest-name*>]	"Copying files between a flash card and a TFTP server" on page 54
copy flash flash monitor standby	"Copying software images between active and standby management modules" on page 53
copy flash flash primary [standby]	"Copying software images between active and standby management modules" on page 53
copy flash flash secondary [standby]	"Copying software images between active and standby management modules" on page 53
copy flash lp <*source-file*> <*dest-file*> <*slot-number*> | all	"Copying files from a management module to an interface module" on page 54
copy flash tftp <*ip-addr*> <*dest-file-name*> primary | secondary	"Copying RX Series IronWare images from flash memory to a TFTP server" on page 54
copy slot1 | slot2 tftp <*ip-addr*> [/<*from-dir-path*>/]<*source-file*> [<*dest-file*>]	"Copying files between a flash card and a TFTP server" on page 54

Commands	See ...
copy tftp slot1 | slot2 <*ip-addr*> [/<*from-dir-path*>/]<*source-file*> <*path-name*> | monitor | primary | secondary	"Copying files between a flash card and a TFTP server" on page 54
copy slot1 | slot2 startup-config [/<*from-dir-path*>/]<*file-name*>	"Copying the startup-config file between a flash card and flash memory" on page 55
copy startup-config slot1 | slot2 [/<*to-dir-path*>/]<*to-name*>	"Copying the startup-config file between a flash card and flash memory" on page 55
copy startup-config tftp <*ip-addr*> [/<*to-dir-path*>]<*to-name*>	"Copying the startup-config file between flash memory and a TFTP server" on page 55
copy tftp startup-config <*ip-addr*> [/<*from-dir-path*>]<*from-name*>	"Copying the startup-config file between flash memory and a TFTP server" on page 55
copy running-config slot1 | slot2 [/<*to-dir-path*>/]<*to-name*>	"Copying the running-config to a flash card or a TFTP server" on page 56
copy slot1 | slot2 running-config [/<*from-dir-path*>/]<*from-name*>	"Loading a running-config from a flash card or a TFTP server" on page 56
copy tftp running-config <*ip-addr*> [/<*from-dir-path*>/]<*from-name*> [overwrite]	"Loading a running-config from a flash card or a TFTP server" on page 56
cp [<*source-dir-path*>]<*source-file-name*> [<*dest-dir-path*>]<*dest-file-name*>	"Copying files using the cp command" on page 57
boot system slot1 | slot2 [/<*dir-path*>/]<*file-name*>	"Rebooting from the system" on page 58
boot system tftp <*ip-address*> <*file-name*>	"Rebooting from the system" on page 58
boot system flash secondary	"Rebooting from the system" on page 58
boot system slot1 <*file-name*> | slot2 <*file-name*> | flash secondary | tftp <*ip-address*> <*file-name*> | bootp	"Configuring the boot source for future reboots" on page 59
locate startup-config	"Displaying the current location for saving configuration changes" on page 59
locate startup-config [slot1 | slot2 | flash-memory] [/<*dir-path-name*>/]<*file-name*>	"Specifying the location for saving configuration changes" on page 59

SNMP

Commands	See ...
show snmp engineid	"Displaying the engine ID" on page 1007
show snmp group	"Defining an SNMP group" on page 1005
show snmp server	"Displaying the SNMP community strings" on page 1003
show snmp user	"Displaying user information" on page 1008
snmp-server community [0] <*string*> ro | rw [view <*viewname*>] [<*standard-acl-name*> | <*standard-acl-id*> | ipv6 <ipv6-access-list-name>]	"Adding an SNMP community string" on page 1002
snmp-server engineid local <*hex-string*>	"Defining the engine ID" on page 1004

Commands	See ...
snmp-server group <*groupname*> v1 | v2c | v3 auth | noauth | priv [access <*standard-acl-id*>] [read <*viewstring*>] [write <*viewstring*>]	"Displaying SNMP groups" on page 1008
snmp-server user <*name*> <*groupname*> v3 [[access <*standard-acl-id*>] [[encrypted] auth md5 <*md5-password*> | sha <*sha-password*> [priv [encrypted] des <*des-password*>]]]	"Defining an SNMP user account" on page 1006
snmp-server view <*name*> <*mib_tree*> included | excluded	"Defining SNMP views" on page 1009

SSH

Commands	See ...
clear public-key	"Importing authorized public keys into the BigIron RX" on page 871
crypto key generate | zeroize	"Generating a host key pair" on page 869
ip ssh authentication-retries <*number*>	"Setting the number of SSH authentication retries" on page 872
ip ssh idle-time <*minutes*>	"Configuring maximum idle time for SSH sessions" on page 874
ip ssh key-authentication no | yes	"Enabling DSA challenge-response authentication" on page 872
ip ssh password-authentication no | yes	"Deactivating user authentication" on page 873
ip ssh permit-empty-passwd no | yes	"Enabling empty password logins" on page 873
ip ssh port <*number*>	"Setting the SSH port number" on page 873
ip ssh pub-key-file tftp I <tftp-server-ip-addr> <*filename*> [remove]	"Importing authorized public keys into the BigIron RX" on page 871
ip ssh scp disable | enable	"Using secure copy" on page 876
ip ssh source-interface ethernet <*slot/port*> | loopback <*num*> | ve <*num*>	"Designating an interface as the source for all SSH packets" on page 874
ip ssh timeout <*seconds*>	"Setting the SSH login timeout value" on page 874
kill ssh <*connection-id*>	"Displaying SSH connection information" on page 875
show ip client-pub-key [begin<*expression*> | exclude <*expression*> | include <*expression*>]	"Importing authorized public keys into the BigIron RX" on page 871
show ip ssh [begin <*expression*> | exclude <*expression*> | include <*expression*>]	"Displaying SSH connection information" on page 875
show who [begin<*expression*> | exclude<*expression*> | include<*expression*>]	"Displaying SSH connection information" on page 875
ssh access-group <*standard-named-acl*> | <*standard-numbered-acl*>	"Filtering SSH access using ACLs" on page 875

Commands	See ...
ssh no-show-host-keys	"Generating a host key pair" on page 869
ssh show-host-keys	"Generating a host key pair" on page 869

sFlow

Commands	See ...
clear statistics	"Clearing sFlow statistics" on page 1035
sflow destination <*ip-addr*> [<*dest-udp-port*>]	"Specifying the collector" on page 1027
sflow enable	"Enabling sFlow forwarding" on page 1029
sflow forwarding	"Enabling sFlow forwarding" on page 1029
sflow polling-interval <*secs*>	"Changing the polling interval" on page 1027
sflow sample <*num*>	"Changing the default sampling rate" on page 1028 "Changing the sampling rate on a port" on page 1029
show sflow	"Displaying sFlow information" on page 1034

STP

Commands	See ...
show spanning-tree [vlan <*vlan-id*>] | [pvst-mode] | [<*num*>] | [detail [vlan <*vlan-id*> [ethernet <*slot/port*>] [| begin<*expression*> | exclude<*expression*> | include<*expression*>]	"Displaying STP information for an entire device" on page 325
show spanning-tree detail [vlan <*vlan-id*> [ethernet <*slot/port*>]]	"Displaying detailed STP information for each interface" on page 328
spanning-tree [ethernet <*slot/port*>] [forward-delay <*value*>] | [hello-time <*value*>] | [max-age <*value*>] | [priority <*value*>]	"Changing STP bridge parameters" on page 322
spanning-tree ethernet <*portnum*> path-cost <*value*> | priority <*value*> | disable | enable	"Changing STP port parameters" on page 322
spanning-tree single [ethernet <*portnum*> path-cost <*value*> | priority <*value*>]	"Enabling SSTP" on page 331
spanning-tree single [forward-delay <*value*>] [hello-time <*value*>] | [maximum-age <*time*>] | [priority <*value*>]	"Enabling SSTP" on page 331
spanning-tree	"Enabling or disabling STP globally" on page 320 "Enabling or disabling STP on a VLAN" on page 320 "Enabling or disabling STP on a port" on page 320

SysLog messages

Commands	See ...
clear logging [dynamic-buffer | static-buffer]	"Displaying the Syslog configuration" on page 1273 "Clearing the Syslog messages from the local buffer" on page 1281
lp show-portname	"Displaying the interface name in Syslog messages" on page 1280
ip show-service-number-in-log	"Displaying TCP/UDP port numbers in Syslog messages" on page 1281
logging buffered <*level*> | <*num-entries*>	"Disabling logging of a message level" on page 1278 "Changing the number of entries the local buffer can hold" on page 1279
logging console	"Enabling real-time display of Syslog messages" on page 1272
logging facility <*facility-name*>	"Changing the log facility" on page 1279
logging host <*ip-addr*> | <*server-name*>	"Specifying a Syslog server" on page 1277 "Specifying an additional Syslog server" on page 1278
logging on [<*udp-port*>]	"Disabling or re-enabling Syslog" on page 1277
show logging	"Displaying the Syslog configuration" on page 1273
terminal monitor	"Enabling real-time display of Syslog messages" on page 1272

System parameters

Commands	See ...
banner <*delimiting-character*> | [motd <*delimiting-character*>]	"Setting a message of the day banner" on page 124
banner exec_mode <*delimiting-character*>	"Setting a privileged EXEC CLI level banner" on page 125
banner incoming <*delimiting-character*>	"Displaying a message on the console when an incoming Telnet session is detected" on page 125
broadcast limit <*number*>	"Configuring CLI banners" on page 124
clock set <hh:mm:ss> <*mm-dd-yy*> | <*mm-dd-yyyy*>	"Setting the system clock" on page 122
clock summer-time	"Setting the system clock" on page 122
clock timezone gmt gmt | us <*time-zone*>	"Setting the system clock" on page 122
hostname <*string*>	"Entering system administration information" on page 114
ip telnet source-interface ethernet <*portnum*> | loopback <*num*> | ve <*num*>	"Configuring an interface as the source for all Telnet packets" on page 118

Commands	See ...
logging enable user-login	"Disabling the Syslog messages and traps" on page 118
mac-age-time <*age-time*>	"Changing the MAC age time" on page 132
multicast limit <*number*>	"Configuring CLI banners" on page 124
route-only	"Enabling or disabling Layer 2 switching" on page 129
router bgp | dvmrp | ospf | pim | rip | vrrp | vrrpe	"Enabling or disabling routing protocols" on page 126
show default values	"Displaying and modifying system parameter default settings" on page 127
show logging	"Examples of Syslog messages for CLI access" on page 117
show sntp associations	"Specifying a Simple Network Time Protocol (SNTP) server" on page 121
show sntp status	"Specifying a Simple Network Time Protocol (SNTP) server" on page 121
show terminal	"Checking the length of terminal displays" on page 126
snmp-server contact <*string*>	"Entering system administration information" on page 114
snmp-server enable traps <*trap-type*>	"Disabling SNMP traps" on page 116
snmp-server enable traps holddown-time <*secs*>	"Setting the SNMP Trap holddown time" on page 116
snmp-server host <*ip-addr*> [0 | 1] <*string*> [port <*value*>]	"Specifying an SNMP trap receiver" on page 115
snmp-server location <*string*>	"Entering system administration information" on page 114
snmp-server trap-source loopback <*num*> | ethernet <*slot/port*> | ve <*num*>	"Specifying a Single trap source" on page 115
sntp poll-interval <*1-65535*>	"Specifying a Simple Network Time Protocol (SNTP) server" on page 121
sntp server <*ip-addr*> | <*hostname*> [<*version*>]	"Specifying a Simple Network Time Protocol (SNTP) server" on page 121
static-mac-address <*mac-addr*> ethernet <*portnum*> [to <*portnum*> ethernet <*portnum*>] [priority <*number*>] [host-type | router-type | fixed-host]	"Assigning static MAC address entries to priority queues" on page 470
system-max ip-static-route <*num*>	"Displaying and modifying system parameter default settings" on page 127
system-max subnet-per-interface <*num*>	"Displaying and modifying system parameter default settings" on page 127
system-max subnet-per-system <*num*>	"Displaying and modifying system parameter default settings" on page 127
terminal length <*number-of-lines*>	"Configuring terminal display" on page 126
unknown-unicast limit <*number*>	"NP based multicast, broadcast, and unknown-unicast rate limiting" on page 503

Topology

Commands	See ...
master-vlan <*vlan-id*>	"Configuring a topology group" on page 437
member-group <*num*>	"Configuring a topology group" on page 437
member-vlan <*vlan-id*>	"Configuring a topology group" on page 437
show topology-group [<*group-id*>]	"Displaying topology group information" on page 438
topology-group <*group-id*>	"Configuring a topology group" on page 437

LAG

Commands	See ...
lag <*lag-name*> static | dynamic | keep-alive	"Creating a Link Aggregation Group (LAG)" on page 235
ports ethernet <*slot/port*> [to <*slot/port*>] [ethernet <*slot/port*>]	"Creating a Link Aggregation Group (LAG)" on page 235
primary port <*slot/port*>	"Configuring the primary port for a LAG" on page 236
trunk-threshold <*number*>	"Specifying the trunk threshold for a trunk Group" on page 236
lacp-port-priority <*slot/port*> <*number*>	"Configuring LACP port priority" on page 237
lacp-timeout [long | short]	"Configuring an LACP timeout" on page 237
deploy [forced | passive]	"Deploying a LAG" on page 237
acl-mirror-port ethe-port-monitored [slot/port] | named-port-monitored [name]	"Configuring ACL-based mirroring" on page 238
disable ethernet [slot/port] | named [name]	"Disabling ports within a LAG" on page 239
enable ethernet [slot/port] | named [name]	"Enabling ports within a LAG" on page 239
monitor ethe-port-monitored [slot/port] | named-port-monitored [name] ethernet [slot/port] [input | output | both]	"Monitoring an individual LAG port" on page 239
port-name <*text*> ethernet <*slot*>/<*portnum*>	"Assigning a name to a port within a LAG" on page 240
sflow-forwarding ethernet [slot/port] | port-name [text]	"Enabling sFlow forwarding on a port within a LAG" on page 240
sflow-subsampling ethernet [slot/port] | port-name [text] <*num*>	"Setting the sFlow sampling rate for a port within a LAG" on page 240
show lag <*lag-name*> [brief] [deployed] [dynamic] [keep-alive] [static]	"Displaying LAG information" on page 241
show statistics [brief] lag [<*lag-name*>]	"Displaying LAG information" on page 241
threshold <*number*>	"Specifying the trunk threshold for a trunk Group" on page 236

UDLD

Commands	See ...
clear link-keepalive statistics	"Clearing UDLD statistics" on page 278
link-keepalive ethernet <*portnum*> [ethernet <*portnum*>]	"Configuring UDLD" on page 274
link-keepalive interval <*num*>	"Changing the keepalive interval" on page 274
link-keepalive retries <*num*>	"Changing the keepalive retries" on page 275
show interface brief	"Displaying information for all ports" on page 275
show link-keepalive [ethernet <*portnum*>]	"Displaying information for all ports" on page 275 "Displaying information for a single port" on page 277
show link-keepalive ethernet	"Displaying information for all ports" on page 275

VLAN

Commands	See ...
add-vlan <*vlan-id*> [to <*vlan-id*>]	"Configuring a VLAN group" on page 291
aggregated-vlan	"Configuring aggregated VLANs" on page 295
default-vlan-id <*vlan-id*>	"Assigning a different ID to the default VLAN" on page 287
ip-proto | ipv6-proto | ipx-proto | atalk-proto | other-proto [<*protocol-vlan-name*>] [static | exclude ethernet <*slot/port*> [to <*slot/port*>] [router-interface ve <*num*>]	"Configuring protocol-based VLANs" on page 287
multicast-flooding	"Hardware flooding for Layer 2 multicast and broadcast packets" on page 311
priority<*num*>	"Assigning or changing a VLAN priority" on page 286
remove-vlan <*vlan-id*> [to <*vlan-id*>]	"Configuring a VLAN group" on page 291
show vlan [<*vlan-id*> | ethernet <*slot/port*> | detail | | begin <*expression*> | exclude <*expression*> | include <*expression*>]	"Displaying VLAN information" on page 314
show vlan-group [<*group-id*>]	"Displaying VLAN group information" on page 317
system-max vlan <*num*>	"Allocating memory for more VLANs or virtual routing interfaces" on page 311
tagged ethernet [to <*slot/port*> | ethernet <*slot/port*>]	"Configuring a VLAN group" on page 291
tag-type <*num*> [ethernet <*slot/port*> [to <*slot/port*>]]	"Configuring aggregated VLANs" on page 295 "Enabling 802.1Q-in-Q tagging" on page 300 "Enabling 802.1q tag-type translation" on page 305
unknown-unicast-flooding	"Unknown unicast flooding on VLAN ports" on page 312
untagged | tagged ethernet <*slot/port*> [to <*slot/port*> | ethernet <*slot/port*>]	"Configuring port-based VLANs" on page 283 "Configuring aggregated VLANs" on page 295

Commands	See ...
uplink-switch ethernet <*portnum*> [to <*portnum*> | ethernet <*portnum*>]	"Configuring uplink ports within a port-based VLAN" on page 313
vlan <*vlan-id*>	"Configuring port-based VLANs" on page 283
vlan <*vlan-id*> [by port]	"Configuring aggregated VLANs" on page 295
vlan <*vlan-id*> [name <*vlan-name*>]	"Configuring port-based VLANs" on page 283
vlan-group <*num*> vlan <*vlan-id*> to <*vlan-id*>	"Configuring a VLAN group" on page 291

VRRP/VRRPE

Commands	See ...
activate	"Configuring the owner" on page 448 "Configuring parameters specific to VRRPE" on page 450
advertise backup	"Backup hello message state and interval" on page 453
backup [priority <*value*>] [track-priority <*value*>]	"Configuring parameters specific to VRRPE" on page 450 "Configuring parameters specific to VRRPE" on page 450 "Track priority" on page 454
backup-hello-interval <*value*>	"Backup hello message state and interval" on page 453
clear ip vrrp-stat	"Clearing VRRP or VRRPE statistics" on page 461
dead-interval <*value*>	"Dead interval" on page 453
hello-interval <*value*>	"Hello interval" on page 452
ip vrrp auth-type no-auth I simple-text auth <*auth-data*>	"Authentication type" on page 451
ip vrrp vrid <*num*>	"Configuring the owner" on page 448 "Configuring parameters specific to VRRPE" on page 450 "VRRP example" on page 462
ip vrrp-extended vrid <*num*>	"Configuring parameters specific to VRRPE" on page 450 "VRRPE example" on page 463
ip vrrp-extended auth-type no-auth | simple-text-auth <*auth-data*>	"Configuring parameters specific to VRRPE" on page 450
ip-address <*ip-addr*>	"Configuring the owner" on page 448 "VRRP example" on page 462
non-preempt-mode	"Backup preempt" on page 454
owner [track-priority <*value*>]	"Configuring the owner" on page 448 "Track priority" on page 454

Commands	See ...
router vrrp	"Configuring the owner" on page 448 "Configuring a backup" on page 449 "VRRP example" on page 462
router vrrp-extended	"Configuring parameters specific to VRRPE" on page 450
show ip vrrp | vrrp-extended [brief | ethernet <*portnum*> | ve <*num*> | stat]	"Displaying summary information" on page 456 "Displaying detailed information" on page 457 "Displaying statistics" on page 460
track-port ethernet <*portnum*> I ve <*num*>	"Track port" on page 453 "VRRPE example" on page 463
use-vrrp-path	"Suppression of RIP advertisements on backup routers for the backup up interface" on page 452

VSRP

Commands	See ...
backup [priority <*value*>] [track-priority <*value*>]	"Configuring basic VSRP parameters" on page 418 "Changing the backup priority" on page 422 "Changing the default track priority" on page 425
enable | disable	"Configuring basic VSRP parameters" on page 418
include-port ethernet <*portnum*>	"Adding or removing a port from the VRID's VLAN" on page 420
initial-ttl <*num*>	"Changing the Time-To-Live (TTL)" on page 423
ip-address <*ip-addr*>	"Configuring a VRID IP address" on page 420
ip vsrp auth-type no-auth I simple-text auth <*auth-data*>	"Configuring authentication" on page 419
non-preempt-mode	"Disabling or re-enabling backup pre-emption" on page 426
show vsrp [vrid <*num*> | vlan <*vlan-id*>]	"Displaying VRID information" on page 429
show vsrp aware	"Displaying the active interfaces for a VRID" on page 433
track-port ethernet <*portnum*> I ve <*num*> [priority <*num*>]	"Specifying a track port" on page 426
vsrp vrid <*num*>	"Configuring basic VSRP parameters" on page 418