Foundry BigIron RX Series Configuration Guide

2100 Gold Street
P.O. Box 649100
San Jose, CA 95164-9100
Tel 408.586.1700
Fax 408.586.1900

November 2005

Contents

Chapter 1
About This Guide

Introduction

This guide describes how to configure the features in the BigIron RX-Series switches from Foundry Networks. Procedures focus on how to configure the features using the Command Line Interface (CLI).

This guide also describes how to monitor Foundry products using statistics and summary commands.

Audience

This manual is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing concepts. They should be familiar with the following protocols if applicable to their network – IP, RIP, OSPF, BGP4, MBGP, IGMP, PIM, VRRP, and VRRPE. They should also be familiar with IPv4 network protocols.

Nomenclature

This guide uses the following typographical conventions to show information:

Italics	Highlights the title of another publication and occasionally emphasizes a word or phrase.
Bold	Highlights a CLI command.
Bold Italic	Highlights a term that is being defined.
NOTE:	A note emphasizes an important fact or calls your attention to a dependency.
CAUTION:	A caution calls your attention to a possible hazard that can damage equipment.
WARNING:	A warning calls your attention to a possible hazard that can cause injury or death.

List of Publications

The following guides apply to the BigIron RX:

- *Foundry BigIron RX Series Installation and Basic Configuration Guide*. This guide describes the BigIron RX Series Switch from Foundry Networks. It provides procedures for installing the interface modules, power supplies, and other components of the switch. It also provides basic configuration procedures of the software. The guide explains how to perform tasks using the CLI.
- *Management Information Base Reference*. This document contains the Simple Network Management Protocol (SNMP) Management Information Base (MIB) objects that are supported in Foundry devices.

To order additional copies of these manuals, do one of the following:

- Call 1.877.TURBOCALL (887.2622) in the United States or 1.408.586.1881 outside the United States.
- Send email to info@foundrynet.com.

List of Supported Features

Table 1.1 lists the supported features on the BigIron RX Series switches.

Table 1.1: List of Supported Features

Category	Feature Description
System Level Features	
• Denial of Service (DoS) protection	Protection from SYN attacks
	Protection from Smurf attacks
• Management Options	Serial and Telnet access to industry-standard Command Line Interface (CLI)
	Web-based GUI
	SNMP versions 1, 2, and 3
	IronView Network Manager Network Manager.
• Security	AAA Authentication
	Local passwords
	RADIUS
	Secure Shell (SSH) version 2
	Secure Copy (SCP)
	TACACS/TACACS+
	User accounts
	802.1X: MD5-challenge EAP type
	Multi-device port authentication
• CPU protection:	There are no CLI commands for CPU protection. The BigIron RX 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

Table 1.1: List of Supported Features (Continued)

Category	Feature Description
• sFlow	sFLow version 5
Layer 2 Features	
• 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
• 802.1w	Rapid Spanning Tree Protocol (RSTP)
• 802.3ad	Dynamic Link Aggregation on tagged and untagged trunks
• Jumbo packets	Layer 2 jumbo packet support
• MRP	Metro Ring Protocol (MRP) Phase 1 and Phase 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	
• Topology Groups	
• Trunk Groups	
• VLANs	802.1Q tagging Port-based VLANs Super Aggregated VLANs (SAV) Dual-mode VLAN ports
• VSRP	Virtual Switch Redundancy Protocol (VSRP)
• Layer 2 ACLs	Replaces MAC filters
• Layer 2 IGMP Snooping	
Layer 3 Features	
• ACLs	Standard or Extended Only Inbound ACLs supported
• BGP	BGP routes BGP peers BGP dampening
• IP Forwarding	Route table

Table 1.1: List of Supported Features (Continued)

Category	Feature Description
• IP Static entries	Routes ARPs Virtual interfaces Secondary addresses
• IS-IS	
• Multicast Routing	Multicast cache L2 IGMP table DVMRP routes PIM-DM PIM-SM
• OSPF	OSPF routes OSPF adjacencies - Dynamic OFPF LSAs OSPF filtering of advertised routes
• PBR	Policy-Based Routing
• RIP versions 1 and 2	RIP routes
• VRRP and VRRPE	Virtual Router Redundancy Protocol (VRRP) and VRRP Extended (VRRPE)

The following features are not supported in the BigIron RX Series switches software releases 02.2.00 or 02.2.01:

NOTE: Commands for some of the following features may exist in the CLI, but they are not supported.

- AppleTalk
- Dynamic IP Routing
- GVRP
- IPX
- Mirroring across VLANs
- MBGP
- MPLS
- MSDP
- MSDP Mesh Groups
- NAT
- RARP
- IGMPv3
- IPv6 and all protocols related to it

- VLANs
  - Dynamic VLANs
  - Private VLANs
  - VLAN translation
- Source IP Port Security

Chapter 2
Getting Started with the Command Line Interface

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

As with other Foundry devices, you can manage a BigIron RX 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.

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

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

Logging on Through the CLI

Once an IP address is assigned to the BigIron RX'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 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. See the *Foundry 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. For 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.

Here is an example:

```
aaa
access-list
all-client
arp
banner
base-mac-addr
boot
```

some lines omitted for brevity...

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

The software provides the following scrolling options:

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

Line Editing Commands

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

Table 2.1: 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 BigIron RX, 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. For 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 BigIron RX. Once you are at the Global Configuration level, you can enter commands to configure the features in the BigIron RX. 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 Unicast Address Family Level

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

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. See “Using Special Characters in Regular Expressions” on page 2-11 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 Foundry 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 BigIron RX.

```
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. For 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 Foundry 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 2.2: 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 2.2: 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. _100_
[ ]	Square brackets enclose a range of single-character patterns. For example, the following regular expression matches output that contains “1”, “2”, “3”, “4”, or “5”: [1-5] You can use the following expression symbols within the brackets. These symbols are allowed only inside the brackets. • **^ –** The caret matches on any characters ***except*** the ones in the brackets. For example, the following regular expression matches output that does ***not*** contain “1”, “2”, “3”, “4”, or “5”: [^1-5] • **-** The hyphen separates the beginning and ending of a range of characters. A match occurs if any of the characters within the range is present. See the example above.

Table 2.2: Special Characters for Regular Expressions (Continued)

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

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

```
BigIron RX# show ip route bgp | include \*
```

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.

Chapter 3
Securing Access to Management Functions

This chapter explains how to secure access to management functions on the BigIron RX. It contains the following sections:

- "Securing Access Methods" on page 3-2 lists the management access methods available on the BigIron RX and the ways you can secure each one
- "Restricting Remote Access to Management Functions" on page 3-4 explains how to restrict access to management functions from remote sources, including Telnet, the Web management interface, and SNMP
- "Setting Passwords" on page 3-10 explains how to set passwords for Telnet access and management privilege levels
- "Setting Up Local User Accounts" on page 3-13 explains how to define user accounts to regulate who can access management functions.
- "Configuring TACACS/TACACS+ Security" on page 3-16 explains how to configure SNMP read-only and read-write community strings on the BigIron RX.
- "Configuring TACACS/TACACS+ Security" on page 3-16 explains how to configure TACACS/TACACS+ authentication, authorization, and accounting.
- "Configuring RADIUS Security" on page 3-30 explains how to configure RADIUS authentication, authorization, and accounting.
- "Configuring Authentication-Method Lists" on page 3-42 explains how to set the order that authentication methods are consulted when more than one is used with an access method.

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

Securing Access Methods

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

Table 3.1: Ways to secure management access to the BigIron RX

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	3-11
Access to the Privileged EXEC and CONFIG levels of the CLI	Not secured	Establish a password for Telnet access to the CLI	3-10
		Establish passwords for management privilege levels	3-11
		Set up local user accounts	3-13
		Configure TACACS/TACACS+ security	3-16
		Configure RADIUS security	3-30
Telnet access	Not secured	Regulate Telnet access using ACLs	3-4
		Allow Telnet access only from specific IP addresses	3-7
		Allow Telnet access only to clients connected to a specific VLAN	3-8
		Specify the maximum number of login attempts for Telnet access	3-8
		Disable Telnet access	3-9
		Establish a password for Telnet access	3-10
		Establish passwords for privilege levels of the CLI	3-11
		Set up local user accounts	3-13
		Configure TACACS/TACACS+ security	3-16
		Configure RADIUS security	3-30

Table 3.1: Ways to secure management access to the BigIron RX (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
Secure Shell (SSH) access For more information on SSH, see "Configuring Secure Shell" on page 29-1	Not configured	Configure SSH	29-1
		Regulate SSH access using ACLs	3-5
		Allow SSH access only from specific IP addresses	3-7
		Establish passwords for privilege levels of the CLI	3-11
		Set up local user accounts	3-13
		Configure TACACS/TACACS+ security	3-16
		Configure RADIUS security	3-30
Web management access	SNMP read or read-write community strings	Regulate Web management access using ACLs	3-5
		Allow Web management access only from specific IP addresses	3-7
		Allow Web management access only to clients connected to a specific VLAN	3-8
		Disable Web management access	3-9
		Configure SSL security for the Web management interface	3-15
		Set up local user accounts	3-13
		Establish SNMP read or read-write community strings for SNMP versions 1 and 2	34-1
		Establishing user groups for SNMP version 3	34-4
		Configure TACACS/TACACS+ security	3-16
		Configure RADIUS security	3-30
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	3-6
		Allow SNMP access only from specific IP addresses	3-7
		Disable SNMP access	3-10
		Allow SNMP access only to clients connected to a specific VLAN	3-8
		Establish passwords to management levels of the CLI	3-11
		Set up local user accounts	3-13
		Establish SNMP read or read-write community strings	3-16

Table 3.1: Ways to secure management access to the BigIron RX (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
TFTP access	Not secured	Allow TFTP access only to clients connected to a specific VLAN	3-9

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 BigIron RX:

- 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 “Access Control List” 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.

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 BigIron RX

The following is an example of configuring BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX using Telnet, the device prompts you for a username and password. By default, you have up to 4 chances to enter a correct username and password. If you do not enter a correct username or password after 4 attempts, the BigIron RX disconnects the Telnet session.

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

```
BigIron RX(config)# telnet login-retries 5
```

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

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

Restricting Remote Access to the Device to Specific VLAN IDs

You can restrict management access to a BigIron RX 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 Foundry 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 BigIron RX. SNMP is required if you want to manage a BigIron RX using IronView Network Manager.

To disable SNMP management of the device:

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

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. See “Setting a Telnet Password” on page 3-10.
- Access to the Privileged EXEC and CONFIG levels of the CLI can be secured by setting passwords for management privilege levels. See “Setting Passwords for Management Privilege Levels” on page 3-11.

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. See “Setting Up Local User Accounts” on page 3-13.

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 BigIron RX 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 BigIron RX. 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. See "Setting Up Local User Accounts" on page 3-13.

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. See "Configuring Authentication-Method Lists" on page 3-42.

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, see "Recovering from a Lost Password" on page 3-12.

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.

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 Foundry 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 Foundry 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 BigIron RX. 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 BigIron RX 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. See "Setting Passwords for Management Privilege Levels" on page 3-11.

If you configure local user accounts, you also need to configure an authentication-method list for Telnet access, Web management access, and SNMP access. See "Configuring Authentication-Method Lists" on page 3-42.

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

Note About Changing Local User Passwords

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

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

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

The current and previous passwords are stored in the device’s running configuration file in encrypted form. For 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 BigIron RX. 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 BigIron RX
- Importing an RSA certificate and private key file from a client (optional)
- Generating a certificate

Enabling the SSL Server on the BigIron RX

To enable the SSL server on the BigIron RX, 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 BigIron RX 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 Foundry device to create them.

If you want to allow the Foundry device to create the digital certificates, see 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 BigIron RX:

- 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 BigIron RX using TACACS/TACACS+.

The TACACS and TACACS+ protocols define how authentication, authorization, and accounting information is sent between a BigIron RX 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 BigIron RX and the TACACS+ server. TACACS+ allows for arbitrary length and content authentication exchanges, which allow any authentication mechanism to be utilized with the BigIron RX. TACACS+ is extensible to provide for site customization and future development features. The protocol allows the BigIron RX 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 BigIron RX 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+, Foundry recommends that you also configure ***authorization***, in which the BigIron RX 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 BigIron RX to log information on the TACACS+ server when specified events occur on the device.

NOTE: By default, a user logging into the device via 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. See “Entering Privileged EXEC Mode After a Telnet or SSH Login” on page 3-24.

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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX obtains a password prompt from a TACACS+ server.
5. The user is prompted for a password.
6. The user enters a password.
7. The BigIron RX 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 BigIron RX 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 BigIron RX using Telnet, SSH, or the Web management interface
2. The user is authenticated.
3. The BigIron RX 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 BigIron RX.
2. The BigIron RX 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 BigIron RX 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 BigIron RX:
   - 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 BigIron RX checks its configuration to see if the event is one for which TACACS+ accounting is required.
3. If the event requires TACACS+ accounting, the BigIron RX 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 BigIron RX 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 BigIron RX 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 Action	Applicable AAA Operations
User enters the command: [no] aaa accounting system default start-stop <method-list>	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list>
	Command accounting (TACACS+): aaa accounting commands <privilege-level> default start-stop <method-list> System accounting start (TACACS+): aaa accounting system default start-stop <method-list>
User enters other commands	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list>
	Command accounting (TACACS+): aaa accounting commands <privilege-level> default start-stop <method-list>

AAA Security for Commands Pasted Into the Running Configuration

If AAA security is enabled on the BigIron RX, 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 and/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

- You must deploy at least one TACACS/TACACS+ server in your network.
- The BigIron RX 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 Foundry 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. See "Identifying the TACACS/TACACS+ Servers" on page 3-21.
2. Set optional parameters. See "Setting Optional TACACS/TACACS+ Parameters" on page 3-22.
3. Configure authentication-method lists. See "Configuring Authentication-Method Lists for TACACS/TACACS+" on page 3-23.

TACACS+ Configuration Procedure

For TACACS+ configurations, use the following procedure:

1. Identify TACACS+ servers. See "Identifying the TACACS/TACACS+ Servers" on page 3-21.

2. Set optional parameters. See “Setting Optional TACACS/TACACS+ Parameters” on page 3-22.
3. Configure authentication-method lists. See “Configuring Authentication-Method Lists for TACACS/TACACS+” on page 3-23.
4. Optionally configure TACACS+ authorization. See “Configuring TACACS+ Authorization” on page 3-24.
5. Optionally configure TACACS+ accounting. See “Configuring TACACS+ Accounting” on page 3-27.

Identifying the TACACS/TACACS+ Servers

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

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 BigIron RX, 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. (See “Configuring Authentication-Method Lists for TACACS/TACACS+” on page 3-23.) 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 and/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 Foundry device sends to the TACACS+ server when trying to authenticate user access.
- Retransmit interval – This parameter specifies how many times the Foundry 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 Foundry 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 Foundry 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 BigIron RX 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 BigIron RX.

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 BigIron RX, the TACACS+ keys are encrypted. For 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 BigIron RX 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 BigIron RX 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 Foundry 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, see page 3-44 under “Examples of Authentication-Method Lists”.

NOTE: For examples of how to define authentication-method lists for types of authentication other than TACACS/TACACS+, see “Configuring Authentication-Method Lists” on page 3-42.

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 Foundry 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 BigIron RX 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 BigIron RX 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 BigIron RX consults a TACACS+ server to determine the privilege level of the authenticated user.

To configure TACACS+ exec authorization on the BigIron RX, 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 "foundry-privlvl" A-V pair is ignored, and the user is granted Super User access.

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

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

Configuring an Attribute-Value Pair on the TACACS+ Server

During TACACS+ exec authorization, the Foundry 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 BigIron RX 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. For 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. See your TACACS+ documentation for the configuration syntax relevant to your server.

If the foundry-privlvl A-V pair is not present, the BigIron RX extracts the last A-V pair configured for the Exec service that has a numeric value. The BigIron RX uses this A-V pair to determine the user's privilege level. For 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 BigIron RX uses the last one that has a numeric value. However, the BigIron RX 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 BigIron RX associates a value from a non-"foundry-privlvl" A-V pair with a Foundry privilege level.

Table 3.2: 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 BigIron RX 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. For 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 BigIron RX.

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

If configured, command accounting is performed for these commands.

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

NOTE: 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 BigIron RX 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 BigIron RX, 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 BigIron RX 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 BigIron RX. 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 BigIron RX. 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 Foundry 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 BigIron RX.

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. For 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 3.3: 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: opens — Number of times the port was opened for communication with the server closes — Number of times the port was closed normally timeouts — Number of times port was closed due to a timeout errors — Number of times an error occurred while opening the port packets in — Number of packets received from the server packets out — Number of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.

The **show web** command displays the privilege level of Web management interface users. For 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 BigIron RX:

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

NOTE: The BigIron RX does not support RADIUS security for SNMP (IronView Network Manager) access.

RADIUS Authentication, Authorization, and Accounting

When RADIUS **authentication** is implemented, the BigIron RX consults a RADIUS server to verify user names and passwords. You can optionally configure RADIUS **authorization**, in which the BigIron RX 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 BigIron RX to log information on a RADIUS accounting server when specified events occur on the device.

NOTE: By default, a user logging into the device via 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. See “Entering Privileged EXEC Mode After a Telnet or SSH Login” on page 3-37.

RADIUS Authentication

When RADIUS authentication takes place, the following events occur:

1. A user attempts to gain access to the BigIron RX 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 BigIron RX sends a RADIUS Access-Request packet containing the username and password to the RADIUS server.
5. The RADIUS server validates the BigIron RX 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 BigIron RX, authenticating the user. Within the Access-Accept packet are three Foundry 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 BigIron RX. 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 BigIron RX.
2. The BigIron RX 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 BigIron RX 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 BigIron RX:
   - 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 BigIron RX checks its configuration to see if the event is one for which RADIUS accounting is required.
3. If the event requires RADIUS accounting, the BigIron RX 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 BigIron RX 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 Action	Applicable AAA Operations
User enters the command: [no] aaa accounting system default start-stop <method-list>	Command authorization: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list> System accounting start: aaa accounting system default start-stop <method-list>
User enters other commands	Command authorization: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list>

AAA Security for Commands Pasted Into the running configuration

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

When you paste commands into the running configuration, and AAA command authorization and/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

- You must deploy at least one RADIUS server in your network.
- The BigIron RX 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 Foundry 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 Foundry vendor-specific attributes on the RADIUS server. See "Configuring Foundry-Specific Attributes on the RADIUS Server" on page 3-34.
2. Identify the RADIUS server to the BigIron RX. See "Identifying the RADIUS Server to the BigIron RX" on page 3-35.

3. Set RADIUS parameters. See “Setting RADIUS Parameters” on page 3-36.
4. Configure authentication-method lists. See “Configuring Authentication-Method Lists for RADIUS” on page 3-37.
5. Optionally configure RADIUS authorization. See “Configuring RADIUS Authorization” on page 3-38.
6. Optionally configure RADIUS accounting. “Configuring RADIUS Accounting” on page 3-39.

Configuring Foundry-Specific Attributes on the RADIUS Server

NOTE: For the BigIron RX, 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 BigIron RX, authenticating the user. Within the Access-Accept packet are three Foundry 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 Foundry 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 BigIron RX.

Foundry’s Vendor-ID is 1991, with Vendor-Type 1. The following table describes the Foundry vendor-specific attributes.

Table 3.4: Foundry 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 3.4: Foundry vendor-specific attributes for RADIUS

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.

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 BigIron RX. For 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 and/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 BigIron RX sends to the RADIUS server when trying to authenticate user access.
- Retransmit interval – This parameter specifies how many times the BigIron RX 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 BigIron RX 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 BigIron RX 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.

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. For 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 Foundry software will retransmit the request up to the maximum number of retransmissions configured. The default retransmit value is 3 retries. The range of retransmit values is from 1 – 5.

To set the RADIUS retransmit limit:

```
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, see page 3-44 under “Examples of Authentication-Method Lists”.

NOTE: For examples of how to define authentication-method lists for types of authentication other than RADIUS, see “Configuring Authentication-Method Lists” on page 3-42.

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 BigIron RX 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 BigIron RX 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 BigIron RX 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 and/or the **aaa authentication login privilege-mode** command exist in the configuration.

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

To configure RADIUS exec authorization on the BigIron RX, enter the following command:

```
BigIron RX(config)# aaa authorization login default radius
BigIron RX(config)# aaa authorization enable default radius
BigIron RX(config)# aaa authorization login privilege-mode
BigIron RX(config)# aaa authorization exec default radius
```

Syntax: aaa authorization exec default radius | none

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

NOTE: If the **aaa authorization exec default radius** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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:

```
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.

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 BigIron RX supports RADIUS accounting for recording information about user activity and system events. When you configure RADIUS accounting on BigIron RX, 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 BigIron RX, 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 BigIron RX 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 BigIron RX. 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 BigIron RX. 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 BigIron RX 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 BigIron RX.

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. For 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 3.5: Output of the show aaa command for RADIUS

Field	Description
Radius key	The setting configured with the **radius-server key** command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Radius retries	The setting configured with the **radius-server retransmit** command.
Radius timeout	The setting configured with the **radius-server timeout** command.
Radius dead-time	The setting configured with the **radius-server dead-time** command.
Radius Server	For each RADIUS server, the IP address, and the following statistics are displayed: Auth Port — RADIUS authentication port number (default 1645) Acct Port — RADIUS accounting port number (default 1646) opens — Number of times the port was opened for communication with the server closes — Number of times the port was closed normally timeouts — Number of times port was closed due to a timeout errors — Number of times an error occurred while opening the port packets in — Number of packets received from the server packets out — Number of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.

The **show web** command displays the privilege level of Web management interface users. For 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. See “Using ACLs to Restrict Remote Access” on page 3-4 or “Restricting Remote Access to the Device to Specific IP Addresses” on page 3-7.

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

- 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. See “Configuring TACACS/TACACS+ Security” on page 3-16.
- 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 Foundry 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 3.6.

Table 3.6: 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. See “Setting a Telnet Password” on page 3-10.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password…** command. See “Setting Passwords for Management Privilege Levels” on page 3-11.
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. See “Configuring a Local User Account” on page 3-14.
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 4
Configuring Basic Parameters

This chapter describes how to configure the following basic system parameters:

- System name, contact, and location – see “Entering System Administration Information” on page 4-2.
- SNMP trap receiver, trap source address, and other parameters – see “Configuring Simple Network Management (SNMP) Traps” on page 4-2.
- Single source address for all Telnet packets – see “Configuring an Interface as the Source for All Telnet Packets” on page 4-6.
- Single source address for all TFTP packets – see “Configuring an Interface as the Source for All TFTP Packets” on page 4-6.
- System time – see “Specifying a Simple Network Time Protocol (SNTP) Server” on page 4-7 and “Setting the System Clock” on page 4-9.
- Broadcast, multicast, or unknown-unicast limits, to support slower third-party devices – see “Limiting Broadcast, Multicast, or Unknown-Unicast Rates” on page 4-10.
- Banners that are displayed on users’ terminals – see “Configuring CLI Banners” on page 4-11.
- Terminal display length – see “Configuring Terminal Display” on page 4-12.
- System defaults and table sizes – see “Displaying and Modifying System Parameter Default Settings” on page 4-13.
- Layer 2 switching – see “Enabling or Disabling Layer 2 Switching” on page 4-16
- MAC age time – see “Changing the MAC Age Time” on page 4-17
- Static MAC address entries – “Configuring Static MAC Addresses” on page 4-17
- Static ARP entries – “Configuring Static ARP Entries” on page 4-18

NOTE: For information about the Syslog buffer and messages, see “Using Syslog” on page A-1.

The BigIron RX is configured with default parameters to allow you to begin using the basic features of the system immediately. However, many advanced features, such as VLANs or routing protocols for the router, must first be enabled at the system (global) level before they can be configured.

You can find system level parameters at the Global CONFIG level of the CLI.

NOTE: Before assigning or modifying any router parameters, you must assign the IP subnet (interface) addresses for each port.

Entering System Administration Information

You can configure a system name, contact, and location for the BigIron RX 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 (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 BigIron RX 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, see “Securing Access to Management Functions” on page 3-1.

Specifying an SNMP Trap Receiver

You can specify a trap receiver to ensure that all SNMP traps sent by the BigIron RX 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 BigIron RX sends all the SNMP traps to the specified host(s) and includes the specified community string. Administrators can therefore filter for traps from a BigIron RX 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 BigIron RX. 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX. Regardless of the port the BigIron RX 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 BigIron RX 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 BigIron RX might not be able to reach the servers, in which case the messages are lost.

By default, the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 Foundry traps is available in the Foundry *Management Information Base Guide*.

Disabling Syslog Messages and Traps for CLI Access

The BigIron RX 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 BigIron RX. 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 BigIron RX 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 BigIron RX.

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 BigIron RX.

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 BigIron RX.

```
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 BigIron RX 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 BigIron RX'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.

Specifying a Simple Network Time Protocol (SNTP) Server

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

NOTE: The BigIron RX 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, Foundry Networks 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 BigIron RX, 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 BigIron RX polls its SNTP server every 30 minutes (1800 seconds). To configure the BigIron RX 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 4.1: 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 4.2: 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

Table 4.2: Output from the show sntp status command (Continued)

This Field...	Indicates...
peer dispersion	Dispersion of the synchronized peer

Setting the System Clock

In addition to SNTP support, the BigIron RX 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, see “Specifying a Simple Network Time Protocol (SNTP) Server” on page 4-7.

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 BigIron RX 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 BigIron RX to adjust the time for any one-hour offset from GMT or for one of the following U.S. time zones:

- US Pacific (default)
- Alaska
- Aleutian
- Arizona
- Central
- East-Indiana
- Eastern
- Hawaii
- Michigan
- Mountain
- Pacific
- Samoa

The default is US Pacific.

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

```
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.

Limiting Broadcast, Multicast, or Unknown-Unicast Rates

The BigIron RX can forward all traffic at wire speed. However, some third-party networking devices cannot handle high forwarding rates for broadcast, multicast, or unknown-unicast packets.

The limits are individually configurable for broadcasts, multicasts, and unknown-unicasts. You can configure limits globally and on individual ports. The valid range is 1 – 4294967295 packets per second. The default is 0, which disables limiting.

NOTE: By default, IP Multicast (including IGMP) is disabled. You can enable it using the **ip multicast passive** | **active** command. As long as IP Multicast is enabled (regardless of whether it is passive or active), no IP Multicast packets (not even IGMP packets) are limited. See “Configuring IP Multicast Traffic Reduction” on page 23-1.

Limiting Broadcasts

To globally limit the number of broadcast packets a BigIron RX forwards to 100,000 per second, enter the following command at the global CONFIG level of the CLI:

```
BigIron RX(config)# broadcast limit 100000
BigIron RX(config)# write memory
```

To limit the number of broadcast packets sent on port 1/3 to 80,000, enter the following commands:

```
BigIron RX(config)# int ethernet 1/3
BigIron RX(config-if-e10000-1/3)# broadcast limit 80000
BigIron RX(config-if-e10000-1/3)# write memory
```

Syntax: broadcast limit <number>

Limiting Multicasts

To globally limit the number of multicast packets a BigIron RX forwards to 120,000 per second, enter the following command at the global CONFIG level of the CLI:

```
BigIron RX(config)# multicast limit 120000
BigIron RX(config)# write memory
```

To limit the number of multicast packets sent on port 3/6 to 55,000, enter the following commands:

```
BigIron RX(config)# int ethernet 3/6
BigIron RX(config-if-e10000-3/6)# multicast limit 55000
BigIron RX(config-if-e10000-3/6)# write memory
```

Syntax: multicast limit <number>

NOTE: The multicast limit is configured at the global level, but the value you enter applies to each management module (slot) installed on the device.

Limiting Unknown Unicasts

To globally limit the number of unknown unicast packets a BigIron RX forwards to 110,000 per second, enter the following command at the global CONFIG level of the CLI:

```
BigIron RX(config)# unknown-unicast limit 110000
BigIron RX(config)# write memory
```


To limit the number of unknown unicast packets sent on port 4/2 to 40,000, enter the following commands:

```
BigIron RX(config)# int ethernet 4/2
BigIron RX(config-if-e10000-4/2)# unknown-unicast limit 40000
BigIron RX(config-if-e10000-4/2)# write memory
```

Syntax: unknown-unicast limit <number>

NOTE: Only the **unknown-unicast limit** is configured on the global level, but the value you enter applies to each management module (slot) installed on the device.

Configuring CLI Banners

The BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron!" when a Telnet CLI session is established:

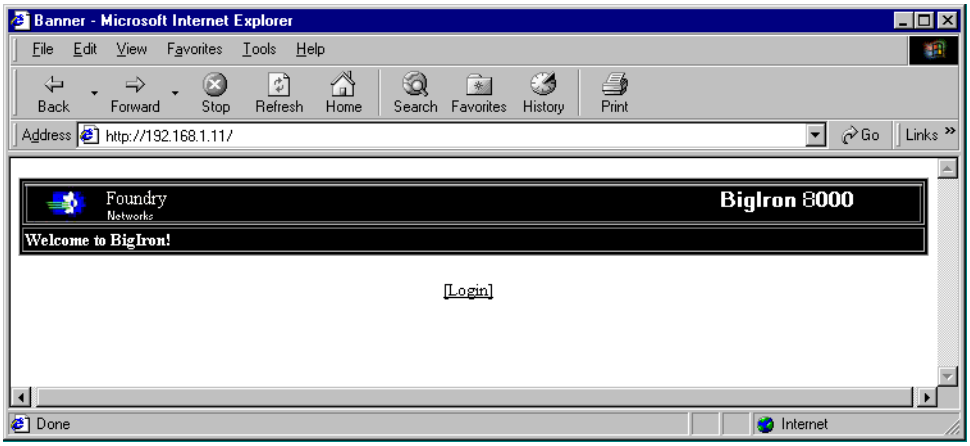
```
BigIron RX(config)# banner motd $ (Press Return)
Enter TEXT message, End with the character '$'.
Welcome to BigIron!! $
```

A delimiting character is established on the first line of the **banner motd** command. You begin and end the message with this delimiting character. The delimiting character can be any character except " (double-quotation mark) and cannot appear in the banner text. In this example, the delimiting character is $ (dollar sign). The text in between the dollar signs is the contents of the banner. The banner text can be up to 2048 characters long and can consist of multiple lines. To remove the banner, enter the **no banner motd** command.

Syntax: [no] banner <delimiting-character> | [motd <delimiting-character>]

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

When you access the Web management interface, the banner is displayed:



Setting a Privileged EXEC CLI Level Banner

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

For 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 BigIron RX supports the following protocols:

- BGP4
- DVMRP
- FSRP
- IP
- OSPF
- PIM
- RIP
- VRRP
- VRRPE

By default, IP routing is enabled on the BigIron RX. 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 BigIron RX, 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 | fsrp | ospf | pim | rip | vrrp | vrrpe

Displaying and Modifying System Parameter Default Settings

The BigIron RX 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 BigIron RX 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 BigIron RX, 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      32768
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       4096       4096
mgmt-port-acl-size   20         100        20
l2-acl-table-entries 64         256        64
vlan-multicast-flood 0          4095       0
```

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 BigIron RX 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:

```
NetIron(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, Foundry BigIron RX 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 Foundry Networks for information.

To globally disable Layer 2 switching on the BigIron RX, 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
```

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 MAC Addresses

You can assign static MAC addresses to the BigIron RX.

NOTE: The BigIron RX also supports the assignment of static IP Routes, and static ARP entries. For details on configuring these types of static entries, see the "Configuring Static Routes" and "Creating Static ARP Entries" sections in the "Configuring IP" chapter.

You can manually input the MAC address of a device to prevent it from being aged out of the system address table, to prevent traffic for a specific device, such as a server, from flooding the network with traffic when it is down, and to assign higher priorities to specific MAC addresses.

You can specify port priority (QoS) and VLAN membership (VLAN ID) for the MAC Address as well as specify device type of either router or host.

The default and maximum configurable MAC table sizes can differ depending on the device. To determine the default and maximum MAC table sizes for your device, display the system parameter values. See "Displaying and Modifying System Parameter Default Settings" on page 4-13.

The BigIron RX can have up to 16,000 static and dynamic MAC address entries stored in the CAM. The ability of the CAM to store depends on the following:

- The number of source MAC address being learned by the CAM.
- The number of destination MAC addresses being forwarded by the CAM
- The distribution of the MAC address entries across ports. For example, it one port is learning all the source MAC addresses, the available of the CAM for that port will be depleted.

Also, a large number of MAC address entries in the MAC table could increase CPU utilization. To alleviate the load on the CPU, use this feature with the Control Plane Security option.

EXAMPLE:

To add a static entry for a server with a MAC address of 1145.5563.67FF and a priority of 7 to port 2 of module 1 of a BigIron RX:

```
BigIron RX(config)# static-mac-address 1145.5563.67FF e 1/2 priority 7
```

Syntax: [no] static-mac-address <mac-addr> ethernet <portnum> [to <portnum> ethernet <portnum>] [priority <number>] [host-type | router-type | fixed-host]

The priority can be 0 – 7. The default priority is 0 or normal-priority.

The default type is host-type.

NOTE: The location of the **static-mac-address** command in the CLI depends on whether you configure port-based VLANs on the device. If the device does not have more than one port-based VLAN (VLAN 1, which is the default VLAN that contains all the ports), the **static-mac-address** command is at the global CONFIG level of the CLI. If the device has more than one port-based VLAN, then the **static-mac-address** command is not available at the global CONFIG level. In this case, the command is available at the configuration level for each port-based VLAN.

Configuring Static ARP Entries

When you create a static ARP entry, the BigIron RX automatically creates a static MAC entry.

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

For more information, see “Configuring ARP Parameters” on page 18-23 and “Creating Static ARP Entries” on page 18-25.

Chapter 5
Configuring Interface Parameters

This chapter describes how to configure the following interface parameters:

- Name – see “Assigning a Port Name” on page 5-2
- IP address – see “Assigning an IP Address to a Port” on page 5-2
- Speed – see “Modifying Port Speed” on page 5-2
- Mode (half-duplex or full-duplex) – see “Modifying Port Mode” on page 5-3
- Status – see “Disabling or Re-Enabling a Port” on page 5-3
- Flow control – see “Disabling or Re-Enabling Flow Control” on page 5-5
- Gigabit negotiate mode – see “Changing the 802.3x Gigabit Negotiation Mode” on page 5-3
- QoS priority – see “Modifying Port Priority (QoS)” on page 5-6
- “Locking a Port to Restrict Addresses” on page 5-5
- “Assigning a Mirror Port and Monitor Ports” on page 5-6
- “Monitoring an Individual Trunk Port” on page 5-7
- “Monitoring 802.3ad Aggregate Links” on page 5-8
- “Mirror Ports for Policy-Based Routing (PBR) Traffic” on page 5-9
- “Displaying Mirror and Monitor Port Configuration” on page 5-10
- “Enabling WAN PHY Mode Support” on page 5-11

Other interface parameters are discussed in the remaining chapters of this manual.

NOTE: To modify Layer 2, Layer 3, or Layer 4 features on a port, see the appropriate section in this chapter or other chapters. For example, to modify Spanning Tree Protocol (STP) parameters for a port, see “Changing STP Port Parameters” on page 10-4.

To configure trunk groups or dynamic link aggregation, see “Configuring Trunk Groups” on page 6-1.

All BigIron RX ports are pre-configured with default values that allow the device to be fully operational at initial startup without any additional configuration. However, in some cases, changes to the port parameters may be necessary to adjust to attached devices or other network requirements.

Assigning a Port Name

A port name can be assigned to help identify interfaces on the network. You can assign a port name to physical ports, virtual routing interfaces, and loopback interfaces.

To assign a name to a port:

```
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 BigIron RX. 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
```

Modifying Port Speed

Each of the 10/100/1000BaseTX ports is designed to auto-sense and auto-negotiate the speed and mode of the connected device. If the attached device does not support this operation, you can manually enter the port speed to operate at either 10 Mbps or 100 Mbps. The default value is 10 or 100 half- or full-duplex.

NOTE: Modifying the port speed of a port that has a pre-configured rate limit policy may result in the inability to remove the port's rate limit policy.

To change the port speed of interface 1/8 from the default of 10/100 auto-sense to 10 Mbps operating at full-duplex, enter the following:

```
BigIron RX(config)# interface e 1/8
BigIron RX(config-if-e10000-1/8)# speed-duplex 10-full
```

Syntax: speed-duplex <value>

The <value> can be one of the following:

- 10-full
- 10-half
- 100-full
- 100-half
- 1000-full
- 1000-half

- auto

The default is auto.

Modifying Port Mode

You can configure a port to accept either full-duplex (bi-directional) or half-duplex (uni-directional) traffic. Port duplex mode and port speed are modified by the same command.

To change the port speed of interface 1/8 from the default of 10/100 auto-sense to 10 Mbps operating at full-duplex, enter the following:

```
BigIron RX(config)# interface e 1/8
BigIron RX(config-if-e10000-1/8)# speed-duplex 10-full
```

Syntax: speed-duplex <value>

The <value> can be one of the following:

- 10-full
- 10-half
- 100-full
- 100-half
- 1000-full
- 1000-half
- auto

The default is auto.

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 BigIron RX, 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 802.3x Gigabit Negotiation Mode

The globally configured Gigabit negotiation mode for 802.3x flow control is the default mode for all Gigabit ports. You can override the globally configured default and set individual ports to the following:

- Negotiate-full-auto – The port first tries to perform a handshake with the other port to exchange capability information. If the other port does not respond to the handshake attempt, the port uses the manually configured configuration information (or the defaults if an administrator has not set the information). This is the default.
- Auto-Gigabit – The port tries to perform a handshake with the other port to exchange capability information.
- Negotiation-off – The port does not try to perform a handshake. Instead, the port uses configuration information manually configured by an administrator.

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 auto-gig
```

This command overrides the global setting and sets the negotiation mode to auto-Gigabit for ports 4/1 – 4/4.

Syntax: gig-default neg-full-auto | auto-gig | neg-off

Changing the Default Gigabit Negotiation Mode

You can configure the default Gigabit negotiation mode to be one of the following:

- Negotiate-full-auto – The port first tries to perform a handshake with the other port to exchange capability information. If the other port does not respond to the handshake attempt, the port uses the manually configured configuration information (or the defaults if an administrator has not set the information). This is the default for Chassis devices.
- Auto-Gigabit – The port tries to perform a handshake with the other port to exchange capability information.
- Negotiation-off – The port does not try to perform a handshake. Instead, the port uses configuration information manually configured by an administrator.

Although the standard for 100BaseTX ports provides an option for a negotiating port to link with a non-negotiating port, the 802.3x standard for Gigabit ports does not provide this option. As a result, unless the ports at both ends of a Gigabit Ethernet link use the same mode (either auto-Gigabit or negotiation-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.

BigIron RX software provides a solution by changing the default negotiation behavior for Gigabit Ethernet ports on the BigIron RX. The new default behavior allows a port to establish a link with another port whether the other port is configured for auto-Gigabit or negotiation-off. By default, Gigabit Ethernet ports first attempt auto-Gigabit. If auto-Gigabit does not succeed (typically because the port at the other end is not configured for auto-Gigabit), the port switches to negotiation-off.

Changing the Negotiation Mode

You can change the negotiation mode globally and for individual ports.

To change the mode globally, enter a command such as the following:

```
BigIron RX(config)# gig-default neg-off
```

This command changes the global setting to negotiation-off. The global setting applies to all Gigabit Ethernet ports except those for which you set a different negotiation mode on the port level.

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 auto-gig
```

This command overrides the global setting and sets the negotiation mode to auto-Gigabit for ports 4/1 – 4/4.

Here is the syntax for globally changing the negotiation mode.

Syntax: gig-default neg-full-auto | auto-gig | neg-off

Here is the syntax for changing the negotiation mode on individual ports.

gig-default neg-full-auto | auto-gig | neg-off

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 BigIron RX 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 BigIron RX, 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 BigIron RX 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 BigIron RX 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>]

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, see “Configuring Quality of Service” on page 16-1.

Assigning a Mirror Port and Monitor Ports

You can monitor traffic on Foundry 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:

PPCR	Port Numbers
1	1 – 12
2	13 – 24

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.
- A port that has sFlow enabled cannot be enable for port monitoring; however, that port can be configured as a mirror port.

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.

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.

The syntax for the examples above are:

Syntax: mirror-port ethernet <slot>/<portnum>

Enter the slot and port number of the port that will be the mirrored.

Syntax: monitor ethernet <slot>/<portnum> both | input | output

The **monitor** command is available at the interface level. Enter the slot and port number of the port that will serve as the monitor port. This port cannot be the same as the mirror port.

Specify **input** if the port will monitor incoming traffic, **output** to monitor outgoing traffic, or **both** to monitor both types of traffic.

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.

Monitoring 802.3ad Aggregate Links

You can monitor 802.3ad aggregate links, as well as individual ports within 802.3ad aggregate links.

This feature is supported on any port that can be configured with 802.3ad link aggregation.

NOTE: The terms ***802.3ad aggregate link*** and ***dynamic trunk group*** are used interchangeably in this section and mean the same thing.

Configuring Port Monitoring on 802.3ad Aggregate Links

By default, when you enable monitoring on the primary port of an 802.3ad aggregate link, the device copies the traffic for all the ports in the dynamic trunk group to the mirror port.

To monitor all of the ports in an 802.3ad aggregate link, enter commands such as the following on the primary port of the dynamic trunk group:

```
BigIron RX(config)# interface e1/1
BigIron RX(config-if-e100-1/1)# link-aggregate monitor ethernet-port-monitored e 1/
1 e 1/10 both
```

These commands enable monitoring of the entire dynamic trunk group and copy both incoming and outgoing traffic to port 1/10, the assigned mirror port. Note that the mirror port (in this case, port 1/10) must already be configured as a mirror port.

Syntax: link-aggregate monitor ethernet-port-monitored ethernet <monitor slot/port> <mirror slot/port> both | in | out

The <monitor slot/port> parameter specifies the port to monitor.

The <mirror slot/port> parameter specifies the port that will receive copies of the monitored port's traffic.

The **both** | **in** | **out** parameter specifies the traffic direction to monitor. There is no default.

Configuring Port Monitoring on an Individual Port in an 802.3ad Aggregate Link

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

```
BigIron RX(config)#interface e1/1
BigIron RX(config-if-e100-1/1)# link-aggregate config-ind-monitor
BigIron RX(config-if-e100-1/1)# link-aggregate monitor ethernet-port-monitored
ethernet 1/1 ethernet 1/10 in
```

Syntax: [no] link-aggregate config-ind-monitor

Syntax: link-aggregate monitor ethernet-port-monitored ethernet <monitor slot/port> <mirror slot/port> in | out | both

The **link-aggregate config-ind-monitor** command enables configuration of individual ports in the dynamic trunk group. Enter this command only once in a dynamic trunk group configuration. After you enter this command, all applicable port configuration commands apply to individual ports only.

NOTE: If you enter **no link-aggregate config-ind-monitor**, the device removes all monitor configuration commands from the individual ports and applies the primary port's configuration to all the ports. Also, once you enter the **no link-aggregate config-ind-monitor** command, any monitor configuration command you enter thereafter applies to the entire trunk group.

The **link-aggregate monitor ethernet-port-monitored ethernet** command in this example enables monitoring of inbound traffic on port 1/1.

- The <monitor slot/port> parameter specifies the port to monitor.
- The <mirror slot/port> parameter specifies the port that will receive copies of the monitored port's traffic.
- The **in** | **out** | **both** parameter specifies the traffic direction to monitor. There is no default.

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 port(s) 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 BigIron RX.

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 6
Configuring Trunk Groups

This chapter describes how to configure trunk groups and 802.3ad link aggregation.

- Trunk groups are manually-configured aggregate links containing multiple ports.
- 802.3ad link aggregation is a protocol that dynamically creates and manages trunk groups.

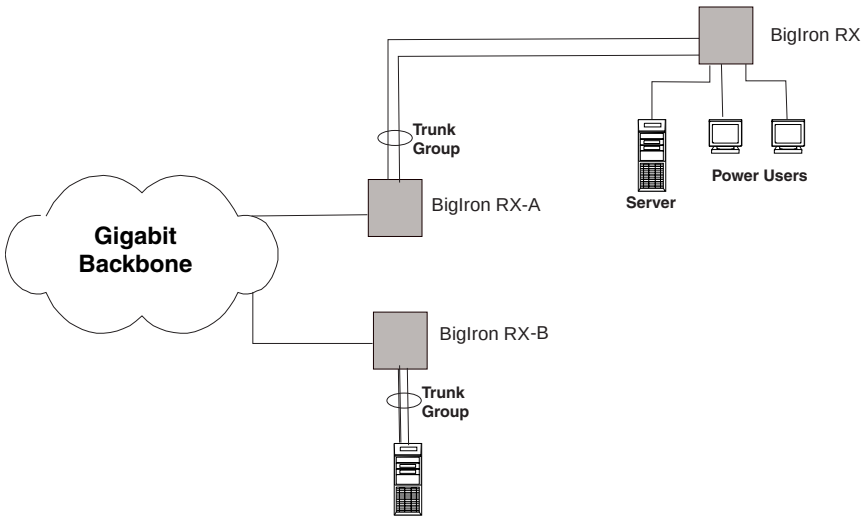
NOTE: You can use both types of trunking on the same device. However, you can use only one type of trunking for a given port. For example, you can configure port 1/1 as a member of a static trunk group or you can enable 802.3ad link aggregation on the port, but you cannot do both.

Trunk groups are manually-configured aggregate links containing multiple ports. Trunk groups enable load sharing of traffic, and they also provide redundant, alternate paths for traffic if any of the segments fail.

You can configure up to 8 ports as a trunk group, supporting transfer rates of up to 8 Gbps of bi-directional traffic. The ports in a trunk group make a single logical link. Therefore, all the ports in a trunk group must be connected to the same device at the other end

Figure 6.1 shows an example of a configuration that uses trunk groups.

Figure 6.1 Trunk Group application within BigIron RX devices

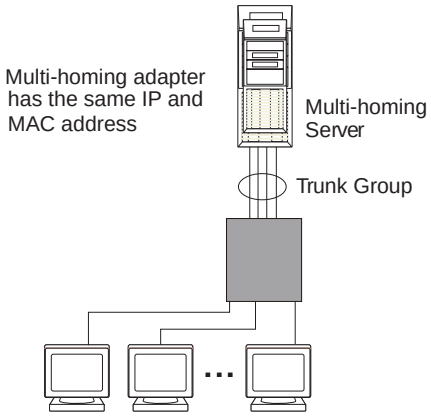


NOTE: The ports in a trunk group make a single logical link. Therefore, all the ports in a trunk group must be connected to the same device at the other end.

Trunk Group Connectivity to a Server

To support termination of a trunk group, the server must have either multiple network interface cards (NICs) or either a dual or quad interface card installed. The trunk server is designated as a server with multiple adapters or a single adapter with multiple ports that share the same MAC and IP address. Figure 6.2 shows an example of a trunk group between a server and BigIron RX.

Figure 6.2 Trunk group between a server and a BigIron RX



Trunk Group Rules

- You cannot configure a port as a member of a trunk group if 802.3ad link aggregation is enabled on the port.
- You can configure up to 31 trunks on a BigIron RX.
- You cannot combine 10/100 ports and Gigabit ports in the same trunk group.
- You cannot combine Gigabit and 10-Gigabit ports in the same trunk group.
- Ports can be in only one trunk group. For example, ports 1/4 cannot be in the Trunk Group 1 and Trunk Group 2.
- 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 trunk group member properties must match the lead port of the trunk group 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 trunk group.
- Make sure the device on the other end of the trunk link can support the same number of ports in the link. For example, if you configure a five-port trunk group on the FastIron Edge Switch switch and the other end is a different type of switch, make sure the other switch can support a five-port trunk group.

Figure 6.3 shows an example of a valid 2-port trunk group links between devices. Ports in a valid 2-port trunk group on one device are connected to two ports in a valid 2-port trunk group on another device. The same rules apply to 4-port trunk groups.

Figure 6.3 Examples of 2-port trunk groups

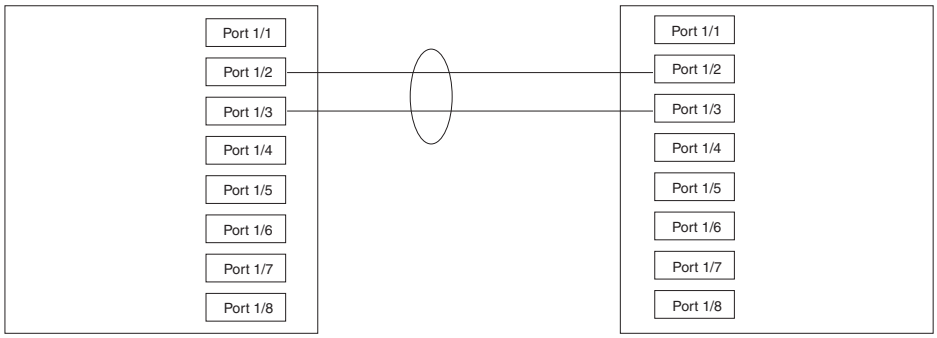
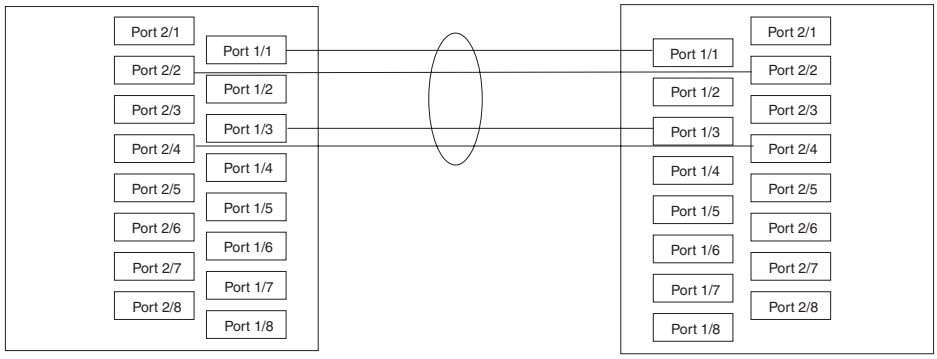


Figure 6.4 shows examples of two devices connected by multi-slot trunk groups.

Figure 6.4 Examples of multi-slot trunk groups



Specifying a Minimum Number of Ports for a Trunk Group

You can configure the BigIron RX 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.

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)# trunk e 3/31 to 3/34
BigIron RX(config-trunk-3/31-3/34)# 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: [no] threshold <number>

You can specify a threshold from 1 (the default) up to the number of ports in the trunk group.

Trunk Formation Rules

The following rules for trunk formation apply to the BigIron RX software release 02.2.01 and later:

- Trunks can be formed from any number of ports, as long as they contain at a minimum of 2 ports and a maximum of ports.

- Ports in a trunk must have the same speed, the same negotiation mode, and the same QoS priority; otherwise, the trunk is rejected.
- Rate Limiting and PBR requirements:

  Primary port policy will apply to all secondary ports. No trunk is rejected.
- Mirroring/Monitoring requirements:

  The trunk is rejected if any trunk port has mirroring or monitoring configured.
- VLAN and inner-VLAN translation:

  The trunk is rejected if any trunk 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 mrp 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.
- You can have a maximum of 31 trunks.

Trunk Group Load Sharing

The BigIron RX shares the traffic load evenly across the ports in the trunk group, while ensuring that packets in the flow are not reordered. To select a port in a trunk group where traffic will be forwarded, BigIron RX calculates a hash index as follows:

- For L2 traffic, the hash index is based on MAC source and destination addresses
- For L3 traffic, the hash index is based is based on the following:
  - IPv4 non-TCP/UDP packets: destination MAC address and source MAC address, source IP address and destination IP address
  - IPv4 TCP packets: destination MAC address and source MAC address, source IP address and destination IP address, and TCP source port and TCP destination port.
  - IPv4 UDP packets: destination MAC address and source MAC address, source IP address and destination IP address, and UDP source port and UDP destination port.

The BigIron RX 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

Configuring a Trunk Group

Use the following procedure when configuring trunk groups:

1. Disconnect the cables from those ports on both systems that will be connected by the trunk group. Do not configure the trunk groups with the cables connected.

   NOTE: If you connect the cables before configuring the trunk groups and then rebooting, the traffic on the ports can create a spanning tree loop.

2. Configure the trunk group on one of the two BigIron RX devices involved in the configuration. See the CLI commands below.
3. Save the configuration changes to the startup-config file.
4. If the device at the other end of the trunk group is another BigIron RX devices, repeat Steps 2 – 4 for the other device. If it is not, refer to the user guide for that device.
5. When the trunk groups on both devices are operational, reconnect the cables to those ports that are now configured as trunk groups, starting with the first port (lead port) of each trunk group.
6. To verify the link is operational, use the **show trunk** command.

Naming a Trunk Port

To name an individual port in a trunk group, enter a command such as the following at the trunk group configuration level:

```
BigIron RX(config-trunk-4/1-4/4)# port-name customer1 ethernet 4/2
```

Syntax: [no] port-name <text> ethernet <slot>/<portnum>

The <text> parameter specifies the port name. The name can be up to 50 characters long.

This command assigns the name “customer1” to port 4/2 in the trunk group consisting of ports 4/1 – 4/4.

Disabling or Re-Enabling a Trunk Port

You can disable or re-enable individual ports in a trunk group. To disable an individual port in a trunk group, enter commands such as the following at the trunk group configuration level:

```
BigIron RX(config-trunk-4/1-4/4)# config-trunk-ind
BigIron RX(config-trunk-4/1-4/4)# disable ethernet 4/2
```

Syntax: [no] config-trunk-ind

Syntax: [no] disable ethernet <slot>/<portnum>

The **config-trunk-ind** command enables configuration of individual ports in the trunk group. If you do not use this command, the **disable** command will be valid only for the primary port in the trunk group and will disable all 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 **disable** command disables the port. The states of other ports in the trunk group are not affected.

If you have configured a name for the trunk port, you can specify the port name, as shown in the following example:

```
BigIron RX(config-trunk-4/1-4/4)# config-trunk-ind
BigIron RX(config-trunk-4/1-4/4)# disable customer1
```


Syntax: disable <portname>

To enable an individual port in a trunk group, enter commands such as the following at the trunk group configuration level:

```
BigIron RX(config-trunk-4/1-4/4)# config-trunk-ind
BigIron RX(config-trunk-4/1-4/4)# enable ethernet 4/2
```

Syntax: enable ethernet <slot>/<portnum>

Syntax: enable <portname>

Disabling or Re-Enabling a Range or List of Trunk Ports

To disable a range of ports in a trunk group, enter commands such as the following:

```
BigIron RX(config)# trunk switch ethernet 2/1 to 2/8
BigIron RX(config-trunk-2/1-2/8)# config-trunk-ind
BigIron RX(config-trunk-2/1-2/8)# disable ethernet 2/2 to 2/5
```

This command disables ports 2/2 – 2/5 in trunk group 2/1 – 2/8.

To disable a list of ports, enter a command such as the following:

```
BigIron RX(config-trunk-2/1-2/8)# disable ethernet 2/2 ethernet 2/4 ethernet 2/7
```

This command disables ports 2/2, 2/4, and 2/7 in the trunk group.

You can specify a range and a list on the same command line. For example, to re-enable some trunk ports, enter a command such as the following:

```
BigIron RX(config-trunk-2/1-2/8)# enable ethernet 2/2 to 2/5 ethernet 2/7
```

Syntax: [no] disable ethernet <slot>/<portnum> [to <slot>/<portnum> | ethernet <slot>/<portnum>]

Syntax: [no] enable ethernet <slot>/<portnum> [to <slot>/<portnum> | ethernet <slot>/<portnum>]

The **to** <slot>/<portnum> parameter indicates that you are specifying a range. Specify the lower port number in the range first, then **to**, then the higher port number in the range.

The **ethernet** <slot>/<portnum> parameter specifies an individual port. You can enter this parameter multiple times to specify a list, as shown in the examples above.

Deleting a Trunk Group

To delete a trunk group, use "**no**" in front of the command you used to create the trunk group. For example, to remove one of the trunk groups configured in the examples above, enter the following command:

```
BigIron RX(config)# no trunk ethernet 1/1 to 1/2 ethernet 3/3 to 3/4
```

Syntax: no trunk ethernet <slot>/<portnum> to <slot>/<portnum>

Displaying Trunk Group Configuration Information

To display trunk group information for specific ports, enter a command such as the following:

```
BigIron RX(config)# show trunk ethernet 1/1 to 1/8

Configured trunks:

Trunk ID: 1
Ports_Configured: 8
Primary Port Monitored: Jointly

Ports        1/1      1/2      1/3      1/4      1/5      1/6      1/7      1/8
Port Names   none     none     none     none     none     longna   test     none
Port_Status  enable   enable   enable   enable   disable  disable  enable   enable
Monitor      on       on       off      on       off      off      off      off
Mirror Port  3/3      3/4      N/A      3/5      N/A      N/A      N/A      N/A
Monitor Dir  both     in       N/A      out      N/A      N/A      N/A      N/A

Operational trunks:

Trunk ID: 1
Duplex: Full
Speed: 1G
Tag: No
Priority: level0
Active Ports: 6

Ports           1/1      1/2       1/3      1/4      1/5    1/6     1/7      1/8
Link_Status     active   active    active   active   down   down    active   active
LACP_Status    ready    ready     ready    expired  down   down    ready    ready
Load Sharing
 Mac Address    3        2         2        2        0      0       6        1
 Multicast      4        2         5        2        0      0       2        3
```

Syntax: show trunk ethernet <slot>/<portnum> to <slot>/<portnum>

The display is divided into sections for configured trunks and operational trunks. A configured trunk group is one that has not been activated yet.

Table 6.1 describes the information displayed by the **show trunk** command.

Table 6.1: CLI Trunk Group Information

This Field...	Displays...
Trunk ID	The trunk group number. The software numbers the groups in the display to make the display easy to use.

Table 6.1: CLI Trunk Group Information (Continued)

This Field...	Displays...
Duplex	The mode of the port, which can be one of the following: • None – The link on the primary trunk port is down. • Full – The primary port is running in full-duplex. • Half – The primary port is running in half-duplex. **Note**: This field and the following fields apply only to operational trunk groups.
Speed	The speed set for the port. The value can be one of the following: • None – The link on the primary trunk port is down. • 10 – The port speed is 10 Mbps. • 100 – The port speed is 100 Mbps. • IG – The port speed is 1000 Mbps.
Tag	Indicates whether the ports have 802.1q VLAN tagging. The value can be Yes or No.
Priority	Indicates the Quality of Service (QoS) priority of the ports. The priority can be a value from 0 – 7.
Active Ports	The number of ports in the trunk group that are currently active.
Ports	The ports in the trunk group.
Link_Status	The link status or each port in the trunk group.
LACP_Status	This field appears in software releases 07.6.03 and later. For more information about this feature, see the section “Displaying and Determining the Status of Aggregate Links” on page 7-9. • Ready - The port is functioning normally in the trunk group and is able to transmit and receive LACP packets. • Expired - The time has expired (as determined by timeout values) and the port has shut down because the port on the other side of the link has stopped transmitting packets. • Down - The port’s physical link is down.
Load Sharing	The number of traffic flows currently being load balanced on the trunk ports. All traffic exchanged within the flow is forwarded on the same trunk port. For information about trunk load sharing, see “Trunk Group Load Sharing” on page 6-4.

Chapter 7
Dynamic Link Aggregation

The software supports the IEEE 802.3ad standard for link aggregation. This standard describes the Link Aggregation Control Protocol (LACP), a mechanism for allowing ports on both sides of a redundant link to configure themselves into a trunk link (aggregate link), without the need for manual configuration of the ports into trunk groups.

When you enable link aggregation on a group of Foundry ports, the Foundry ports can negotiate with the ports at the remote ends of the links to establish trunk groups.

Usage Notes

- You cannot use 802.3ad link aggregation on a port configured as a member of a static trunk group.
- When the feature dynamically adds or changes a trunk group, the **show trunk** command displays the trunk as both configured and active. However, the **show running-config** or **write terminal** command does not contain a trunk command defining the new or changed trunk group.
- If the feature places a port into a trunk group as a secondary port, all configuration information except information related to link aggregation is removed from the port. For example, if port 1/3 has an IP interface, and the link aggregation feature places port 1/3 into a trunk group consisting of ports 1/1 – 1/4, the IP interface is removed from the port.
- You can enable link aggregation on 802.1q tagged ports (ports that belong to more than one port-based VLAN).

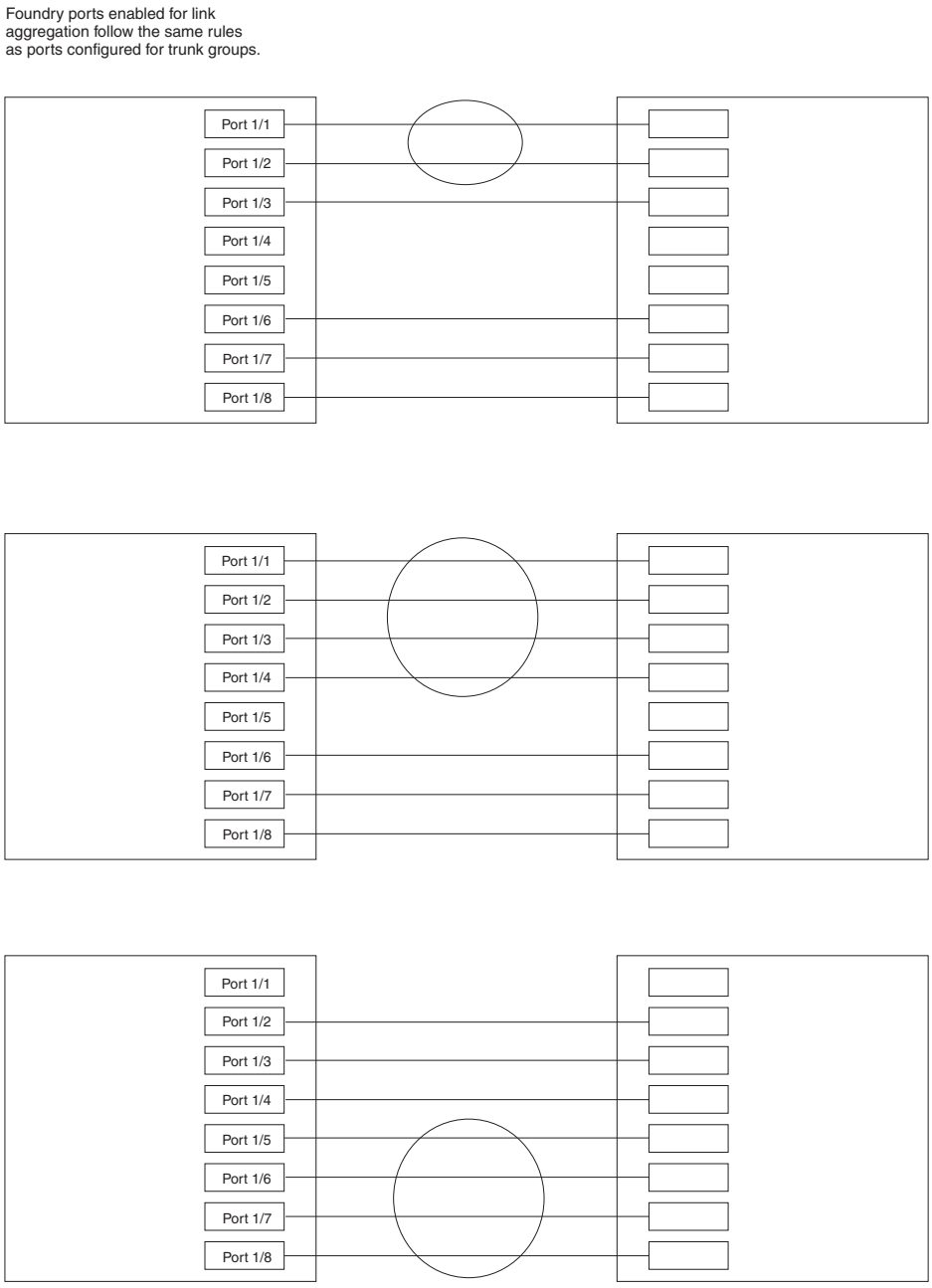
Configuration Rules

Foundry ports follow the same configuration rules for dynamically created aggregate links as they do for statically configured trunk groups. See “Trunk Group Rules” on page 6-2 and “Trunk Group Load Sharing” on page 6-4.

NOTE: Foundry recommends that you disable or remove the cables from the ports you plan to enable for dynamic link aggregation. Doing so prevents the possibility that LACP will use a partial configuration to talk to the other side of a link. A partial configuration does not cause errors, but does sometimes require LACP to be disabled and re-enabled on both sides of the link to ensure that a full configuration is used. It's easier to disable a port or remove its cable first. This applies both for active link aggregation and passive link aggregation.

Figure 7.1 on page 7-2 shows some examples of valid aggregate links.

Figure 7.1 Examples of valid aggregate links



In this example, assume that link aggregation is enabled on all of the links between the Foundry device on the left and the device on the right (which can be either a Foundry device or another vendor's device). Notice that some ports are not able to join an aggregate link even though link aggregation is enabled on them. The ports that are not members of aggregate links in this example are not following the configuration rules for trunk links on Foundry devices.

The Foundry rules apply to a Foundry device even if the device at the other end is from another vendor and uses different rules. See "Trunk Group Rules" on page 6-2.

The link aggregation feature automates trunk configuration but can coexist with Foundry's trunk group feature. Link aggregation parameters do not interfere with trunk group parameters.

NOTE: Use the link aggregation feature only if the device at the other end of the links you want to aggregate also supports IEEE 802.3ad link aggregation. Otherwise, you need to manually configure the trunk links.

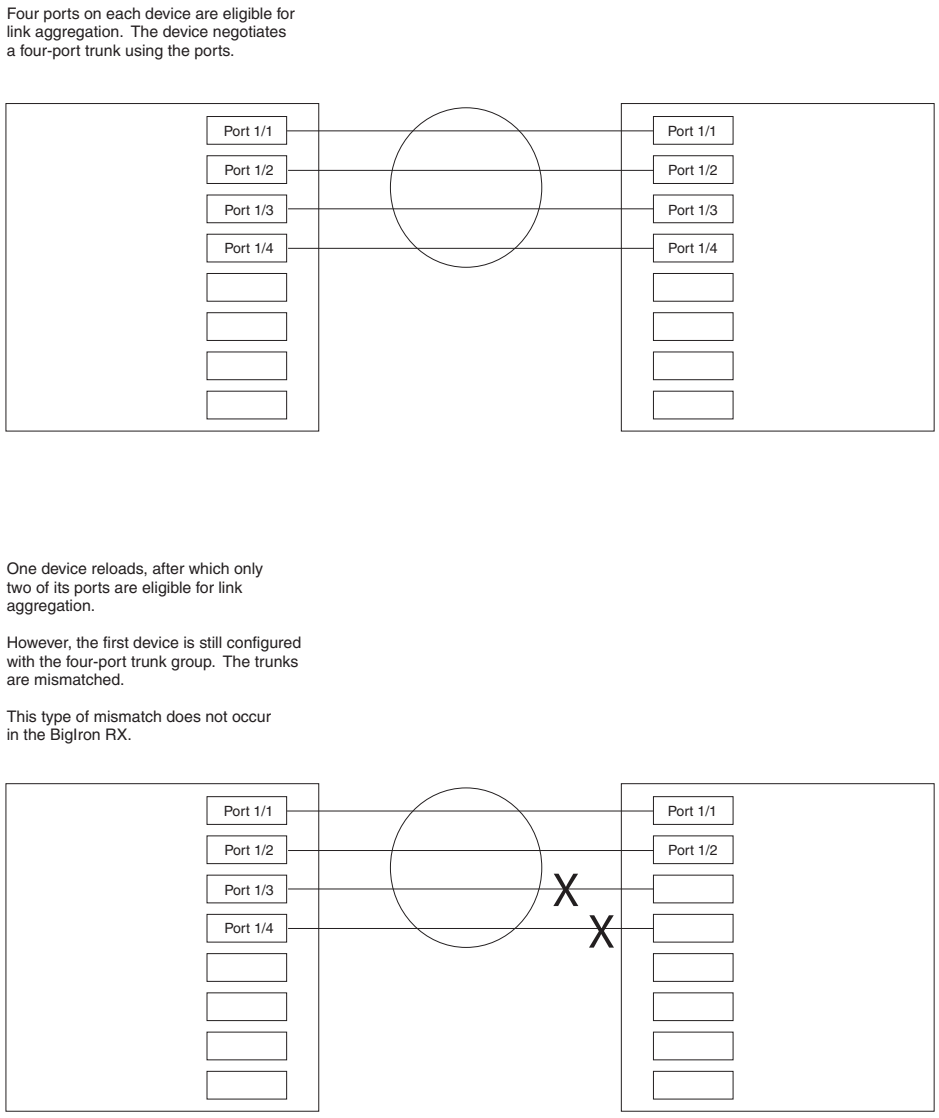
Link aggregation support is disabled by default. You can enable the feature on an individual port basis, in active or passive mode.

- Active mode – When you enable a port for active link aggregation, the Foundry port can exchange standard LACP Protocol Data Unit (LACPDU) messages to negotiate trunk group configuration with the port on the other side of the link. In addition, the Foundry port actively sends LACPDU messages on the link to search for a link aggregation partner at the other end of the link, and can initiate an LACPDU exchange to negotiate link aggregation parameters with an appropriately configured remote port.
- Passive mode – When you enable a port for passive link aggregation, the Foundry port can exchange LACPDU messages with the port at the remote end of the link, but the Foundry port cannot search for a link aggregation port or initiate negotiation of an aggregate link. Thus, the port at the remote end of the link must initiate the LACPDU exchange.

Adaptation to Trunk Disappearance

The Foundry device will tear down an aggregate link if the device at the other end of the link reboots or brings all the links down. Tearing the aggregate link down prevents a mismatch if the other device has a different trunk configuration following the reboot or re-establishment of the links. Figure 7.2 shows an example of a trunk port mismatch.

Figure 7.2 Trunk port mismatch



Adaptation to trunk disappearance prevents trunk mismatches caused when one device changes the number of ports in group of ports that has become part of an 802.3 aggregate link. If a device changes the number of ports in an active aggregate link, the Foundry device on the other end of the link tears down the link. Once the other device recovers, 802.3 can renegotiate the link without a mismatch.

Enabling Link Aggregation

By default, link aggregation is disabled on all ports. To enable link aggregation on a set of ports, enter commands such as the following at the interface configuration level of the CLI.

NOTE: Configuration commands for link aggregation differ depending on whether you are using the default link aggregation key automatically assigned by the software, or if you are assigning a different, unique key. Follow the commands below, according to the type of key you are using. For more information about keys, see "Configuring Keys for Ports" on page 7-6.

Using the Default Key Assigned by the Software

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# link-aggregate active
BigIron RX(config)# interface ethernet 1/2
BigIron RX(config-if-e1000-1/2)# link-aggregate active
```

The commands in this example enable the active mode of link aggregation on ports 1/1 and 1/2. The ports can send and receive LACPDU messages. Note that these ports will use the default key, since one has not been explicitly configured.

Assigning a Unique Key

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# link-aggregate configure key 10000
BigIron RX(config-if-e1000-1/1)# link-aggregate active
BigIron RX(config)# interface ethernet 1/2
BigIron RX(config-if-e1000-1/2)# link-aggregate configure key 10000
BigIron RX(config-if-e1000-1/2)# link-aggregate active
```

The commands in this example assign the key 10000 and enable the active mode of link aggregation on ports 1/1 and 1/2. The ports can send and receive LACPDU messages.

NOTE: As shown in this example, when configuring a key, it is pertinent that you assign the key prior to enabling link aggregation.

The following commands enable passive link aggregation on ports 1/5 – 1/8:

```
BigIron RX(config)# interface ethernet 1/5 to 1/8
BigIron RX(config-if-1/5-1/8)# link-aggregate passive
```

The commands in this example enable the passive mode of link aggregation on ports 1/5 – 1/8. These ports wait for the other end of the link to contact them. After this occurs, the ports can send and receive LACPDU messages.

To disable link aggregation on a port, enter a command such as the following:

```
BigIron RX(config-if-e1000-1/8)# link-aggregate off
```

Syntax: [no] link-aggregate active | passive | off

Syntax: [no] link-aggregate configure [system-priority <num>] | [port-priority <num>] | [key <num>]]

NOTE: For more information about keys, including details about the syntax shown above, see "Configuring Keys for Ports" on page 7-6.

Configuring Link Aggregation Parameters

On a BigIron RX running software release 02.2.01, the **lacp system-priority** command specifies the Foundry device's link aggregation priority relative to the devices at the other ends of the links on which link aggregation is enabled. You can change the value of this parameter globally by entering the following command:


```
BigIron RX(config)# lacp system-priority 3
```

Syntax: lacp system-priority <number>

Specify 1 – 65535 for number. A higher value indicates a lower priority. The default is 1.

NOTE: If you are connecting the Foundry device to another vendor's device and the link aggregation feature is not working, set the system priority on the Foundry device to a lower priority (a higher priority value). In some cases, this change allows the link aggregation feature to operate successfully between the two devices.

You can change the settings for the following link aggregation parameters, on an individual port basis:

- Port priority
- Key

Configuring Port Priority

The port priority determines the active and standby links. When a group of ports is negotiating with a group of ports on another device to establish a trunk group, the Foundry port with the highest priority becomes the default active port. The other ports (with lower priorities) become standby ports in the trunk group.

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(config-if-e1000-1/1)# priority
```

Syntax: priority <number>

You can specify a priority from 0 – 65535. A higher value indicates a lower priority. The default is 1.

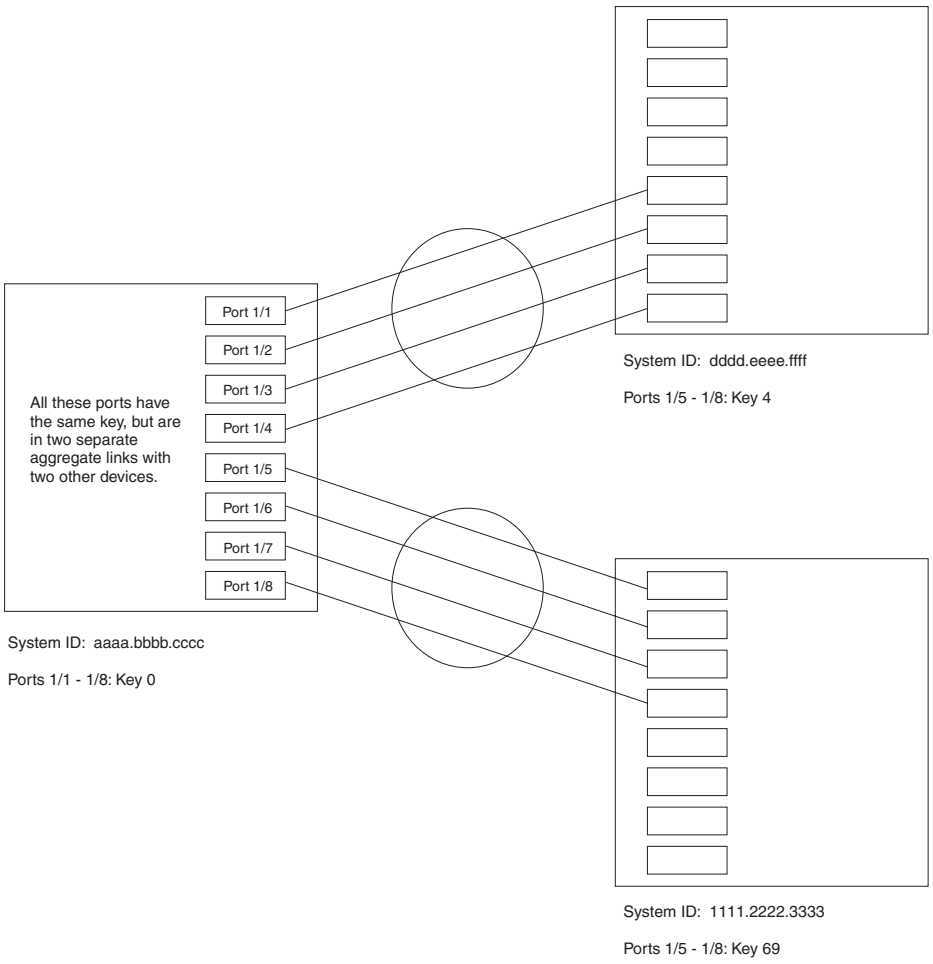
The primary port in the port group becomes the default active port. The primary port is the lowest-numbered port in a valid trunk-port group.

Configuring Keys for Ports

Every port that is 802.3ad-enabled has a key. The key identifies the group of potential trunk ports to which the port belongs. Ports with the same key are called a key group and are eligible to be in the same trunk group. When you enable link-aggregation on a tagged or untagged port, Foundry's software assigns a default key to the port.

All ports within an aggregate link must have the same key. However, if the device has ports that are connected to two different devices, and the port groups allow the ports to form into separate aggregate links with the two devices, then each group of ports can have the same key while belonging to separate aggregate links with different devices. Figure 7.3 on page 7-7 shows an example.

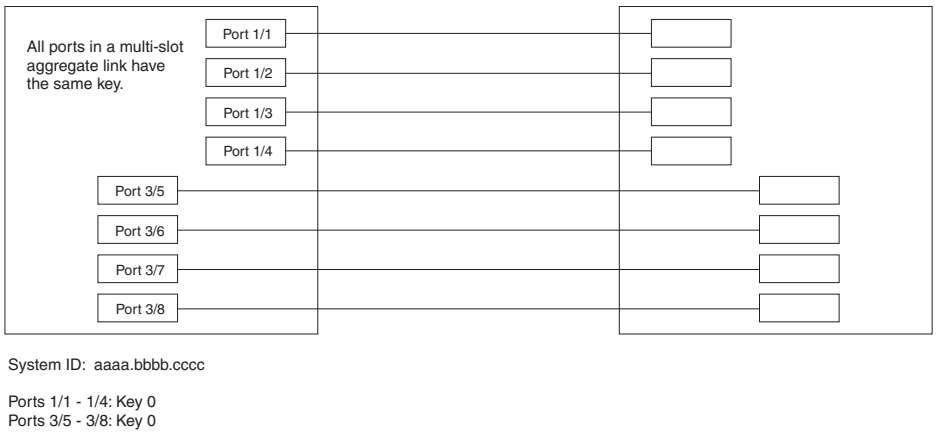
Figure 7.3 Ports with the same key in different aggregate links



Notice that the keys between one device and another do not need to match. The only requirement for key matching is that all the ports within an aggregate link on a given device must have the same key.

Devices that support multi-slot trunk groups can form multi-slot aggregate links using link aggregation. However, the link aggregation keys for the groups of ports on each module must match. For example, if you want to allow link aggregation to form an aggregate link containing ports 1/1 – 1/4 and 3/5 – 3/8, you must change the link aggregation key on one or both groups of ports so that the key is the same on all eight ports. Figure 7.4 on page 7-8 shows an example.

Figure 7.4 Multi-slot aggregate link



By default, the device's ports are divided into 4-port groups. The software dynamically assigns a unique key to each 4-port group. If you need to divide a 4-port group into two 2-port groups, change the key in one of the groups so that the two 2-port groups have different keys. For example, if you plan to use ports 1/1 and 1/2 in VLAN 1, and ports 1/3 and 1/4 in VLAN 2, change the key for ports 1/3 and 1/4.

For key configuration only, configuration commands differ depending on whether or not link aggregation is enabled on the port(s). Follow the appropriate set of commands below, according to your system's configuration.

Configuring Keys For Ports with Link Aggregation Disabled

Use the command sequence below to change the key for ports that do not have link aggregation enabled, and for all other link aggregation parameters (i.e., system priority, port priority, and link type).

```
BigIron RX(config)# interface ethernet 1/1 to 1/4
BigIron RX(config-if-1/1-1/4)# link-aggregate configure key 10000
BigIron RX(config-if-1/1-1/4)# interface ethernet 3/5 to 3/8
BigIron RX(config-if-3/5-3/8)# link-aggregate configure key 10000
```

NOTE: If you change the key for a port group, Foundry recommends that you use the value 10000 or higher, to avoid potential conflicts with dynamically created keys.

Configuring Keys For Ports with Link Aggregation Enabled

As shown in this command sequence, to change the key on ports that already have link aggregation enabled, you must first turn OFF link aggregation, configure the new key, then re-enable link aggregation.

```
BigIron RX(config)# interface ethernet 1/1 to 1/4
BigIron RX(config-if-1/1-1/4)# link-aggregate off
BigIron RX(config-if-1/1-1/4)# link-aggregate configure key 10000
BigIron RX(config-if-1/1-1/4)# link-aggregate active
BigIron RX(config-if-1/1-1/4)# interface ethernet 3/5 to 3/8
BigIron RX(config-if-3/5-3/8)# link-aggregate off
BigIron RX(config-if-3/5-3/8)# link-aggregate configure key 10000
BigIron RX(config-if-3/5-3/8)# link-aggregate active
```

These commands change the key for ports 1/1 – 1/4 and 3/5 – 3/8 to 10000. Since all ports in an aggregate link must have the same key, the command in this example enables ports 1/1 – 1/4 and 3/5 – 3/8 to form a multi-slot aggregate link.

Syntax: [no] link-aggregate configure [port-priority <num>] | [key <num>]

The **port-priority** <num> parameter specifies an individual port's priority within the port group. A higher value indicates a lower priority. You can specify a priority from 0 – 65535. The default is 1.

The **key** <num> parameter identifies the group of ports that are eligible to be aggregated into a trunk group. The software automatically assigns a key to each group of ports. The software assigns the keys in ascending numerical order, beginning with 10000. You can change a port group's key to a value from 10000 – 65535.

NOTE: If you change the key for a port group, Foundry recommends that you use the value 10000 or higher, to avoid potential conflicts with dynamically created keys.

You can enter one or more of the command's parameters on the same command line, in any order.

Viewing Keys for Tagged Ports

To display link aggregation information, including the key for a specific port, enter a command such as the following at any level of the CLI:

```
BigIron RX# show link-aggregation ethernet 1/1

System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp]
1/1        1        1   10000   No    L   No   No   No   No   No   No
```

The command in this example shows the key and other link aggregation information for port 1/1.

To display link aggregation information, including the key for all ports on which link aggregation is enabled, enter the following command at any level of the CLI:

```
BigIron RX# show link-aggregation

System ID: 0004.8055.b200
Long  timeout: 90, default: 90
Short timeout: 3, default: 3

Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1        1        1   10000   Yes   S   Agg  Syn  Col  Dis  Def  No   Dwn
1/2        1        1   10000   Yes   S   Agg  Syn  Col  Dis  Def  No   Dwn
2/1        1        1   10000   Yes   S   Agg  Syn  Col  Dis  Def  No   Dwn
2/2        1        1   10000   Yes   S   Agg  Syn  Col  Dis  Def  No   Dwn
4/1        1        1   10000   Yes   S   Agg  Syn  Col  Dis  Def  No   Dwn
4/2        1        1   10000   Yes   S   Agg  Syn  Col  Dis  Def  No   Dwn
4/3        1        1   10000   Yes   S   Agg  Syn  Col  Dis  Def  No   Dwn
4/4        1        1   48000   Yes   S   Agg  Syn  Col  Dis  Def  No   Dwn
4/17       1        1   48000   Yes   S   Agg  Syn  Col  Dis  Def  No   Ope
4/18       1        1   48000   Yes   S   Agg  Syn  Col  Dis  Def  No   Ope
4/19       1        1   48000   Yes   S   Agg  Syn  Col  Dis  Def  No   Ope
4/20       1        1   48000   Yes   S   Agg  Syn  Col  Dis  Def  No   Ope
```

For information about the fields in this display, see Table 7.1 on page 7-10.

Syntax: show link-aggregation [ethernet <slot>/<portnum>]

Possible values: N/A

Default value: N/A

Displaying and Determining the Status of Aggregate Links

Displaying Link Aggregation and Port Status Information

Use the **show link-aggregation** command to determine the operational status of ports associated with aggregate links.

To display the link aggregation information for a specific port, enter a command such as the following at any level of the CLI:

```
BigIron RX(config-if-1/1-1/8)# show link-aggregation ethernet 1/1
System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp] [Ope]
1/1        1        1    10000   No    L   No   No   No   No   No   No    Ope
```

The command in this example shows the link aggregation information for port 1/1.

To display the link aggregation information for all ports on which link aggregation is enabled, enter the following command at any level of the CLI:

```
BigIron RX(config)# show link-aggregation

System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1        1        1   10000   No    L   Agg  Syn  No   No   Def  Exp  Ope
1/2        1        1   10000   No    L   Agg  Syn  No   No   Def  Exp  Ina
1/3        1        1   10000   No    L   Agg  Syn  No   No   Def  Exp  Ina
1/4        1        1   10000   No    L   Agg  Syn  No   No   Def  Exp  Blo
1/5        1        1   48000   No    L   Agg  No   No   No   Def  Exp  Ope
1/6        1        1   48000   No    L   Agg  No   No   No   Def  Exp  Ope
1/7        1        1   48000   No    L   Agg  No   No   No   Def  Exp  Dwn
1/8        1        1   48000   No    L   Agg  No   No   No   Def  Exp  Dwn
```

Syntax: show link-aggregation [ethernet <slot>/<portnum>]

Use **ethernet <slot>/<portnum>** to display link-aggregation information for a specific port. Ports that are configured as part of an aggregate link must also have the same key.

The **show link aggregation** command shows the following information.

Table 7.1: CLI Display of Link Aggregation Information

This Field...	Displays...
System ID	Lists the base MAC address of the device. This is also the MAC address of port 1 (or 1/1).
Port	Lists the port number.
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.

Table 7.1: CLI Display of Link Aggregation Information (Continued)

This Field...	Displays...
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.
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.

Table 7.1: CLI Display of Link Aggregation Information (Continued)

This Field...	Displays...
Def	Indicates whether the port is using default link aggregation values. The port uses default values if it has not received link aggregation information through LACP from the port at the remote end of the link. This field can have one of the following values: • Def – The port has not received link aggregation values from the port at the other end of the link and is therefore using its default link aggregation LACP settings. • No – The port has received link aggregation information from the port at the other end of the link and is using the settings negotiated with that port.
Exp	Indicates whether the negotiated link aggregation settings have expired. The settings expire if the port does not receive an LACPDU message from the port at the other end of the link before the message timer expires. This field can have one of the following values: • Exp – The link aggregation settings this port negotiated with the port at the other end of the link have expired. The port is now using its default link aggregation settings. • No – The link aggregation values that this port negotiated with the port at the other end of the link have not expired, so the port is still using the negotiated settings.
Ope	• Ope (operational) - The port is operating normally. • Ina (inactive) - The port is inactive because the port on the other side of the link is down or has stopped transmitting LACP packets. • 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.

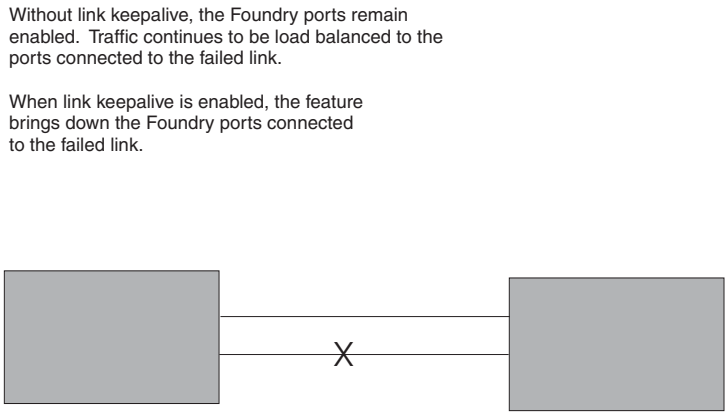
Displaying Trunk Group and LACP Status Information

Use the **show trunk** command to determine the status of LACP. See “Displaying Trunk Group Configuration Information” on page 6-7.

Chapter 8
Configuring Uni-Directional Link Detection (UDLD)

Uni-directional Link Detection (UDLD) monitors a link between two BigIron RX 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 8.1 shows an example.

Figure 8.1 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, the port concludes that the link has failed and takes the port down.

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.

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>]

Table 8.1: 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 BigIron RX 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 BigIron RX port and the BigIron RX 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 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 8.2: CLI Display of Detailed UDLD Information

This Field...	Displays...
Current State	The state of the logical link. This is the link between this BigIron RX port and the BigIron RX 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 BigIron RX.
Remote Port	The port number on the BigIron RX at the remote end of the link.
Local System ID	A unique value that identifies this BigIron RX. The ID can be used by Foundry technical support for troubleshooting.
Remote System ID	A unique value that identifies the BigIron RX 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 Foundry 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 9
Configuring Virtual LANs (VLANs)

This chapter describes how to configure Virtual LANs (VLANs) on a BigIron RX. Topics include:

Types of VLANs

You can configure the following types of VLANs on a BigIron RX:

- Layer 2 port-based VLAN – a set of physical ports that share a common, exclusive Layer 2 broadcast domain
- Layer 3 protocol-based VLANs – a subset of ports within a port-based VLAN that share a common, exclusive broadcast domain for Layer 3 broadcasts of the specified protocol. The following protocol-based VLANs are supported:

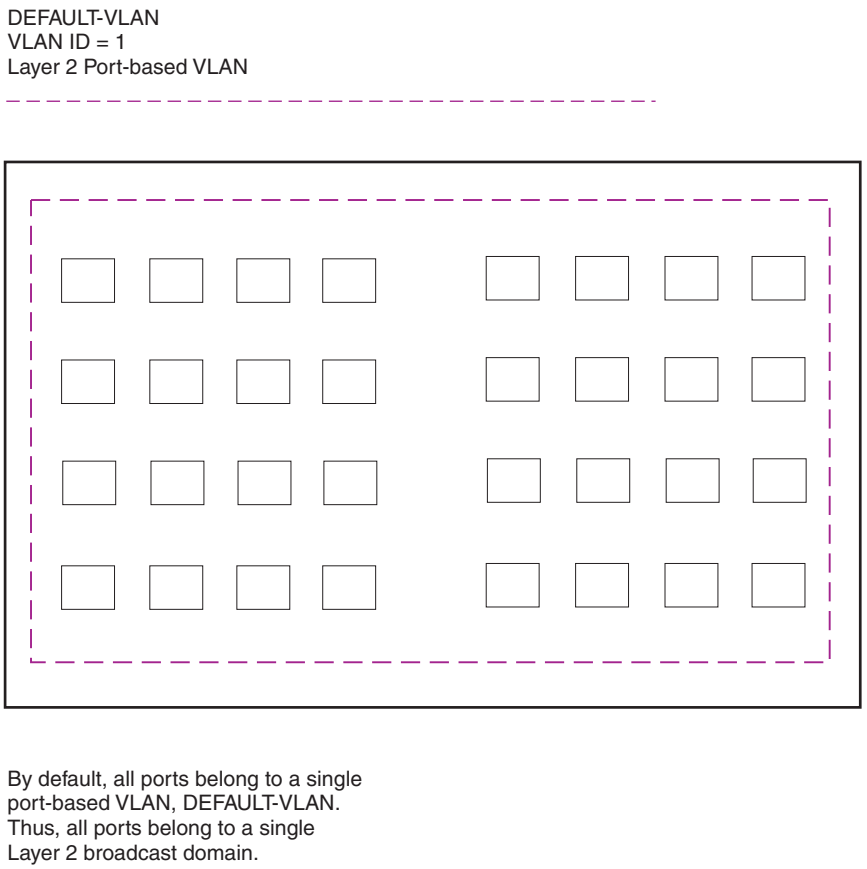
- IP protocol VLAN
- IPv6 protocol VLANs
- IPX protocol VLAN
- AppleTalk protocol VLAN
- VLANs for other protocols

Default VLAN

By default, all the ports on a BigIron RX are in a single port-based VLAN. This VLAN is called DEFAULT-VLAN and is VLAN number 1. The BigIron RX does not contain any protocol VLANs or IP subnet, IPX network, or AppleTalk VLANs by default.

Figure 9.1 shows an example of the default Layer 2 port-based VLAN.

Figure 9.1 Default Layer 2 port-based VLAN



When you configure a port-based VLAN, one of the configuration items you provide is the ports that are in the VLAN. When you configure the VLAN, the BigIron RX automatically removes the ports that you place in the VLAN from DEFAULT-VLAN. By removing the ports from the default VLAN, the BigIron RX ensures that each port resides in only one Layer 2 broadcast domain.

Some network configurations may require that a port be able to reside in two or more Layer 2 broadcast domains (port-based VLANs). In this case, you can enable a port to reside in multiple port-based VLANs by tagging the port. See “IEEE 802.1q Tagging” on page 9-4.

If your network requires that you use VLAN ID 1 for a user-configured VLAN, you can reassign the default VLAN to another valid VLAN ID.

Assigning a Different VLAN ID to the Default VLAN

The default VLAN is not configurable. If you want to use the default VLAN ID 1 as a configurable VLAN, you can assign a different VLAN ID to the default VLAN. For example, enter the following command:

```
BigIron RX(config)# default-vlan-id 4094
```

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 – 4094.

NOTE: Changing the default VLAN name does not change the properties of the default VLAN. Changing the name allows you to use the VLAN ID “1” as a configurable VLAN. Also, VLAN 4092 is a reserved VLAN for BigIron RX.

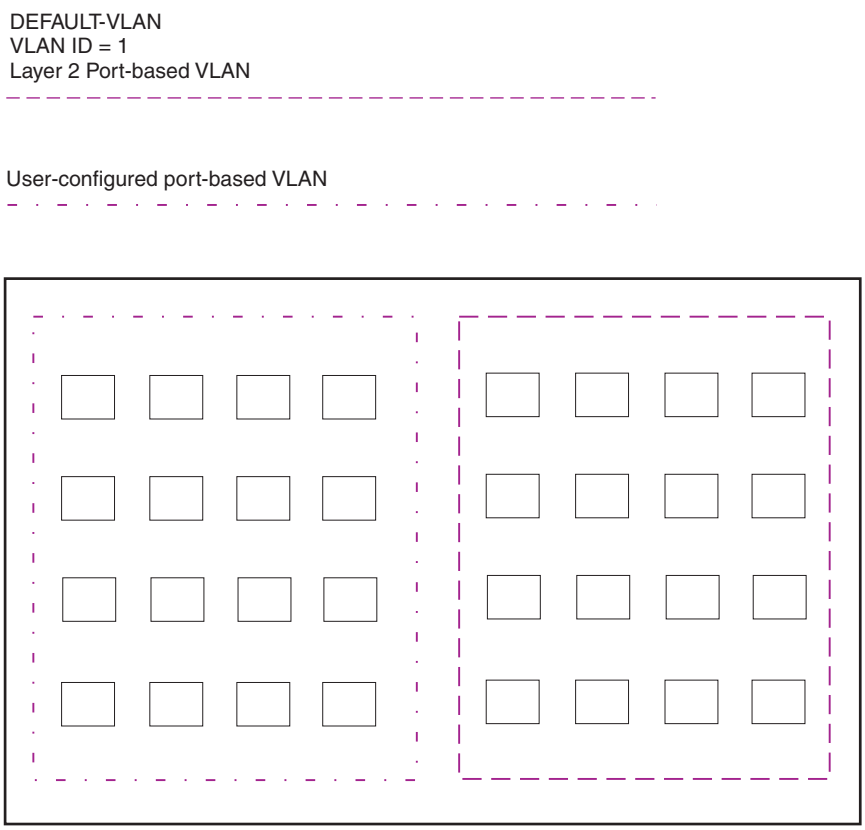
Layer 2 Port-Based VLANs

A port-based VLAN is a subset of ports on a BigIron RX that constitutes a Layer 2 broadcast domain.

By default, all the ports on a BigIron RX are members of the default VLAN. Thus, all the ports on the BigIron RX constitute a single Layer 2 broadcast domain. You can configure multiple port-based VLANs. When you configure a port-based VLAN, the BigIron RX automatically removes the ports you add to the VLAN from the default VLAN.

Figure 9.2 shows an example of a BigIron RX on which a Layer 2 port-based VLAN has been configured.

Figure 9.2 BigIron RX containing user-defined Layer 2 port-based VLAN



A port can belong to only one port-based VLAN, unless you apply 802.1q tagging to the port. ***802.1q tagging*** allows the port to add a four-byte tag field, which contains the VLAN ID, to each packet sent on the port. You also can configure port-based VLANs that span multiple devices by tagging the ports within the VLAN. The tag enables each device that receives the packet to determine the VLAN the packet belongs to. 802.1q tagging applies only to Layer 2 VLANs, not to Layer 3 VLANs.

Since each port-based VLAN is a separate Layer 2 broadcast domain, by default each VLAN runs a separate instance of the Spanning Tree Protocol (STP).

Layer 2 traffic is bridged within a port-based VLAN, and Layer 2 broadcasts are sent to all the ports within the VLAN.

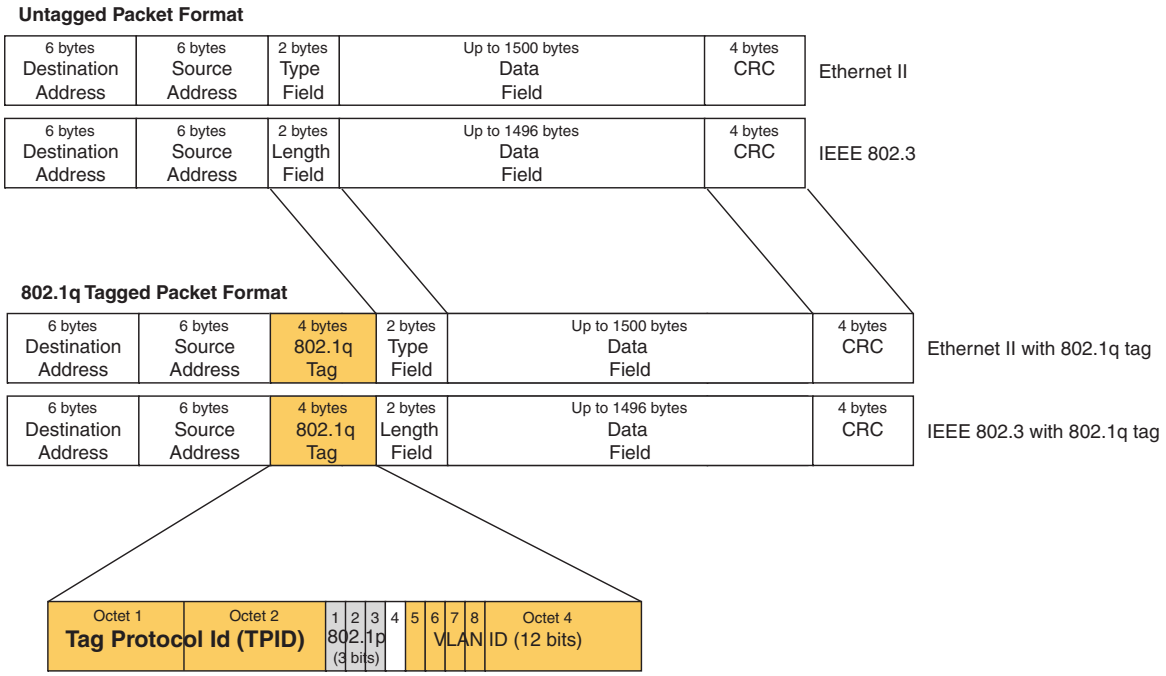
IEEE 802.1q Tagging

802.1q tagging is an IEEE standard that allows a networking device to add information to a Layer 2 packet in order to identify the VLAN membership of the packet. A BigIron RX tag a packet by adding a four-byte tag to the packet. The tag contains the tag value, which identifies the data as a tag, and also contains the VLAN ID of the VLAN from which the packet is sent.

- The default tag value is 8100 (hexadecimal). This value comes from the 802.1q specification. You can change this tag value on a global basis on BigIron RX if needed to be compatible with other vendors' equipment.
- The VLAN ID is determined by the VLAN on which the packet is being forwarded.

Figure 9.3 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 9.3 Packet containing Foundry's 802.1QVLAN tag



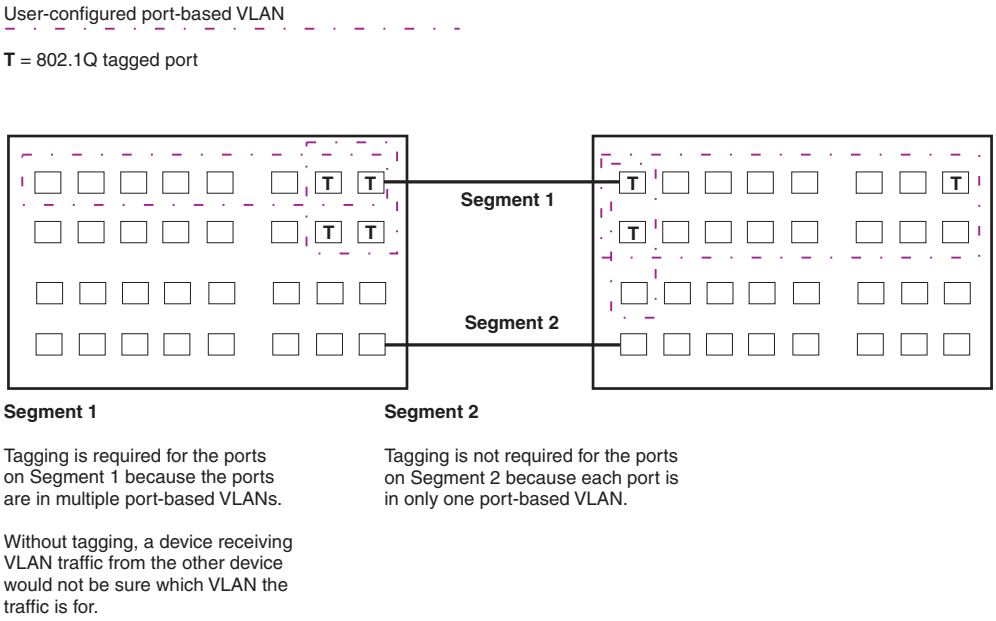
NOTE: You cannot configure a port to be a member of the default port-based VLAN and another port-based VLAN at the same time. Once you add a port to a port-based VLAN, the port is no longer a member of the default VLAN. The port returns to the default VLAN only if you delete the other VLAN(s) that contains the port.

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 BigIron RX switches is compatible with other Foundry devices.

Figure 9.4 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 9.4 VLANs configured across multiple devices



Configuring a Port-Based VLAN

To configure a port-based VLAN, enter commands such as the following:

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# untag e 1/9 to 1/16
```

The commands configure port-based VLAN 2 and add untagged ethernet ports 1/9 through 1/16 to it.

Syntax: vlan <vlan-id> [name <vlan-name>]

Syntax: untagged | tagged ethernet <slot/port> [to <slot/port> | ethernet <slot/port>]

The **untag** command takes ports from the default VLAN and puts them in the port-based VLAN. (The default VLAN contains all the ports in the system by default.) The **untag** command also allows the ports to process packets that do not contain 802.1q tagging.

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
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 <slot>/<portnum> [to <slot>/<portnum> | ethernet <slot>/<portnum>]

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.

Modifying a Port-Based VLAN

You can make the following modifications to a port-based VLAN:

- Add or remove a port.
- Change its priority.
- Enable or disable STP. See “IEEE 802.1D Spanning Tree Protocol (STP)” on page 10-1.

Removing a Port from a Port-Based VLAN

To remove a port from a port-based VLAN, enter the commands such as the following:

```
BigIron RX(config)# vlan 4
BigIron RX(config-vlan-4)# no untag ethernet 1/11
```

The command removes the untagged port 1/11 from VLAN 4 and puts it in the default VLAN.

Syntax: no untagged | tagged ethernet <slot/port> [to <slot/port> | ethernet <slot/port>]

Assigning or Changing a Priority to a VLAN

You can prioritize traffic on a VLAN by assigning a priority to a VLAN.

```
BigIron RX(config-vlan-2)# priority 2
```

Syntax: priority <num>

Possible Values: 0 - 7, "0" assigns the lowest priority and "7", the highest priority. The default is "0".

Removing a Port-Based VLAN

To remove a VLAN, enter a vlan command at the global CONFIG level, such as the following:

```
BigIron RX(config)# no vlan 5
```

Syntax: no vlan <vlan-id>

Layer 3 Protocol-Based VLANs

Protocol-based VLANs provide the ability to define separate broadcast domains for Layer 3 protocols such as IP, IPv6, IPX, AppleTalk, and others, within a single Layer 2 broadcast domain. Some applications for this feature might include security between departments with unique protocol requirements. This feature enables you to limit the amount of broadcast traffic to end-stations, servers, and routers.

If you want some or all of the ports within a port-based VLAN to be organized according to Layer 3 protocol, you must configure a Layer 3 protocol-based VLAN within the port-based VLAN. You can configure the following types of protocol-based VLAN within a port-based VLAN. All the ports in the Layer 3 VLAN must be in the same Layer 2 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.
- IPv6 – a subset of ports in a port-based VLAN that share a common, exclusive network broadcast domain for IPv6 packets The device sends IPv6 broadcasts to all ports within the IPv6 protocol VLAN.
- IPX – The device sends IPX broadcasts to all ports within the IPX protocol VLAN.
- Other – The device sends broadcasts for all protocol types other than those listed above to all ports within the

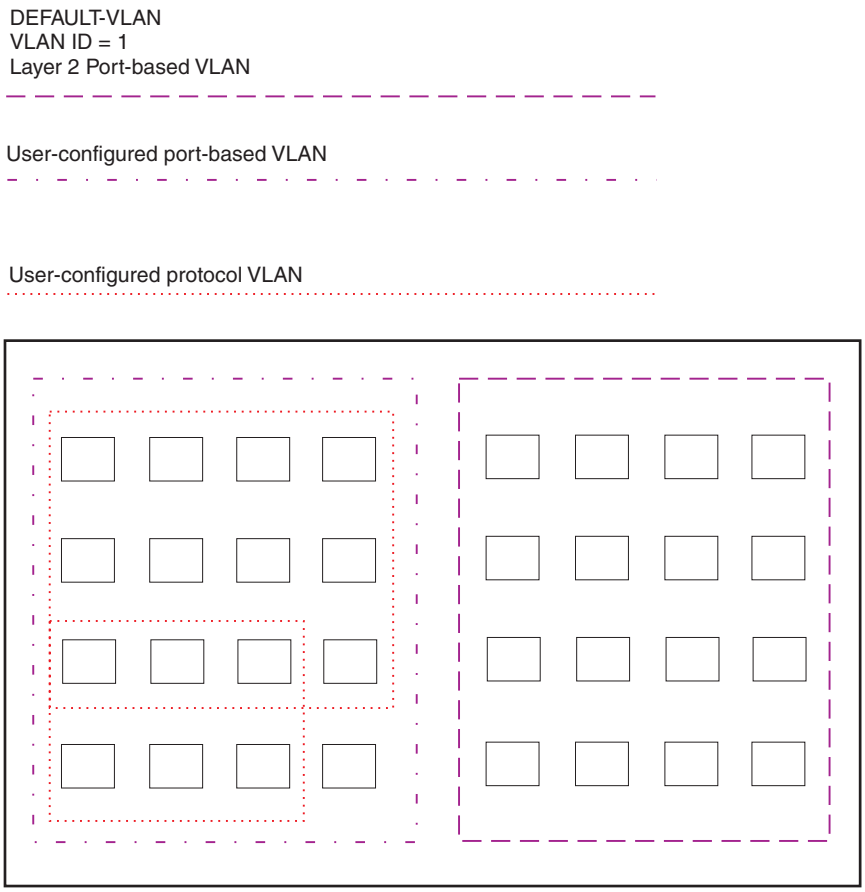
VLAN.

You can configure a protocol-based VLAN as a broadcast domain for IPv6 traffic. When the BigIron RX receives an IPv6 multicast packet (a packet with 06 in the version field and 0xFF as the beginning of the destination address), the BigIron RX forwards the packet to all other ports in the VLAN.

NOTE: The BigIron RX forwards all IPv6 multicast packets to all ports in the VLAN except the port that received the packet, and does not distinguish among subnet directed multicasts.

Figure 9.5 shows a Layer 3 protocol-based VLANs configured within a Layer 2 port-based VLAN.

Figure 9.5 Layer 3 protocol VLANs within a Layer 2 port-based VLAN



Layer 3 protocol-based VLANs cannot span Layer 2 port-based VLANS. However, Layer 3 protocol-based VLANs can overlap within a Layer 2 port-based VLAN.

Static and Excluded Port Membership

Protocol-based VLANs have the following membership types:

- Static ports
- Excluded ports

All ports must be explicitly designated as static ports or excluded from a VLAN.

Static Ports

Static ports are permanent members of a protocol-based VLAN; they never age out. They remain active members of the protocol-based VLAN regardless of whether they receive traffic for the VLAN's protocol.

To add static ports to a protocol-based VLAN, you explicitly identify ports as static when you configure a protocol-based VLAN.

Excluded Ports

To prevent a port in a port-based VLAN from becoming a member of a protocol VLAN, you can explicitly exclude the port when you configure the protocol-based VLAN.

Excluded ports do not leak broadcast packets.

Configuring Protocol-Based VLANs

The following example configures IP protocol, IPv6 protocol, and other-protocol VLANs within a port-based VLAN. Although AppleTalk and IPX protocol VLANs are not part of the example, the configuration steps are alike.

To configure IP, IPv6, and other protocol VLANs, enter commands as shown in the following:

1. Create a port-based VLAN and assign untagged and tagged ports to it.

```
BigIron RX(config)# vlan 2
BigIron RX(config-vlan-2)# untag e 1/1 to 1/8
BigIron RX(config-vlan-2)# tag e 1/24
```

2. Enable STP and set the priority to make the device the root bridge for the port-based VLAN 2.

```
BigIron RX(config-vlan-2)# spanning-tree
BigIron RX(config-vlan-2)# spanning-tree priority 500
```

3. Create the IP and IPv6 protocol VLANs and assign static ports to them.

```
BigIron RX(config-vlan-2)# ip-proto name Gray
BigIron RX(config-vlan-group-ip-proto)# static e 1/1 to 1/4 e 1/24
BigIron RX(config-vlan-group-ip-proto)# exclude e 1/5 to 1/8
BigIron RX(config-vlan-group-ip-proto)# ipv6-proto name Blue
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
```

4. To prevent machines with non-IP and non-IPv6 protocols from getting into VLAN 2, create another protocol VLAN to exclude all other protocols from VLAN 2. To do so, enter the following commands:

```
BigIron RX(config-vlan-group-ipx-proto)# other-proto name Block_other_proto
BigIron RX(config-vlan-group-other-proto)# exclude e 1/1 to 1/8 e 1/24
BigIron RX(config-vlan-group-other-proto)#
```

Syntax: 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>]

The **static** ethernet <slot/port> [to <slot/port>] parameter adds the specified port(s) within the port-based VLAN as static port(s) to the protocol VLAN.

The **exclude** ethernet <slot/port> [to <slot/port>] parameter excludes the specified port(s) from the protocol VLAN.

The **router-interface ve** <num> parameter adds a configured virtual routing interface to the protocol VLAN.

Spanning Tree Protocol (STP) in VLANs

STP is a Layer 2 protocol. Thus, you cannot enable or disable STP for individual protocol VLANs. The STP state of a port-based VLAN determines the STP state for all the Layer 2 broadcasts within the port-based VLAN. This is true even though Layer 3 protocol broadcasts are sent on Layer 2 within the VLAN.

It is possible that STP will block one or more ports in a protocol 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 so long as at least one port in the virtual routing interface's protocol VLAN is not blocked by STP.

If you enable Single STP (SSTP) on the device, the ports in all VLANs on which STP is enabled become members of a single spanning tree. The ports in VLANs on which STP is disabled are excluded from the single spanning tree.

For more information, see "Configuring Spanning Tree Protocol" .

Trunk Group Ports and VLAN Membership

A trunk group is a set of physical ports that are configured to act as a single physical interface. Each trunk group's port configuration is based on the configuration of the lead port, which is the lowest numbered port in the group.

If you add a trunk port to a VLAN, all of the ports in the trunk group become members of that VLAN.

Assigning Trunk Group Ports

When a trunk group port is assigned to a VLAN, all other members of the trunk group are automatically added to that VLAN. See "Trunk Group Rules" on page 6-2 for more information.

Summary of VLAN Configuration Rules

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, IP, IPv6, IPX, AppleTalk, Decnet, and NetBIOS are at the middle level of the hierarchy.

NOTE: You cannot have a protocol-based VLAN and a subnet or network VLAN of the same protocol type in the same port-based VLAN. For example, you can have an IPX protocol VLAN and IP subnet VLAN in the same port-based VLAN, but you cannot have an IP protocol VLAN and an IP subnet VLAN in the same port-based VLAN, nor can you have an IPX protocol VLAN and an IPX network VLAN in the same port-based VLAN.

As a BigIron RX receives packets, the VLAN classification starts from the highest level VLAN first. Therefore, if an interface is configured as a member of both a port-based VLAN and an IP protocol VLAN, IP packets coming into the interface are classified as members of the IP protocol VLAN because that VLAN is higher in the VLAN hierarchy.

When a port of a VLAN on a BigIron RX receives a packet, the device forwards the packet based on the following VLAN hierarchy:

- If the port belongs to an IP subnet VLAN, IPX network VLAN, or AppleTalk protocol VLAN and the packet belongs to the corresponding IP subnet, IPX network, or AppleTalk protocol range, the device forwards the packet to all the ports within that VLAN.
- If the packet is a Layer 3 packet but cannot be forwarded as described above, but the port is a member of a Layer 3 protocol VLAN for the packet's protocol, the device forwards the packet on all the Layer 3 protocol VLAN's ports.
- If the packet cannot be forwarded based on either of the VLAN membership types listed above, 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

- A port can belong to multiple, unique, overlapping Layer 3 protocol-based VLANs.

- 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.
- When both port and protocol-based VLANs are configured on a given device, all protocol VLANs must be strictly contained within a port-based VLAN. A protocol 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.
- IP protocol VLANs and IP subnet VLANs cannot operate concurrently on the system or within the same port-based VLAN.
- One of each type of protocol VLAN is configurable within each port-based VLAN on the BigIron RX.
- Removing a configured port-based VLAN from a BigIron RX automatically removes any protocol-based VLAN, IP subnet VLAN, or IPX network VLAN, or any virtual Ethernet router interfaces defined within the port-based VLAN.

Configuration Considerations

Note the following configuration limitations:

- The dynamic protocol VLAN option is not supported.
- The other-protocol option defines a protocol-based VLAN for protocols that do not require a singular protocol broadcast domain or are not currently supported on the Foundry device. It is used as a catch-all rule to mean all other protocols in addition to those already assigned.

  For example, in the following VLAN configuration, IP protocol is defined and the "other-proto" option is set to become operational when a non-IPv4 packet is received.

```
BigIron RX(config)#vlan 5
BigIron RX(config-vlan-5)#ip-proto
BigIron RX(config-vlan-5)#other-proto
```

Configuration Examples of Port-Based and Protocol-Based VLANs

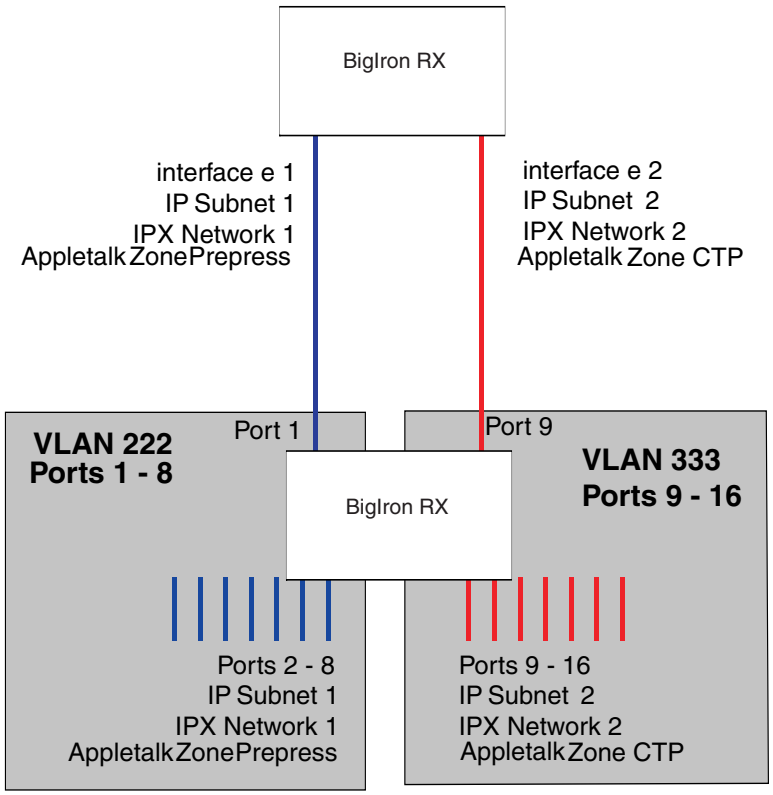
Configuring Port-Based VLANs

Port-based VLANs allow you to provide separate spanning tree protocol (STP) domains or broadcast domains on a port-by-port basis.

EXAMPLE:

Figure 9.6 shows a simple port-based VLAN configuration using a single BigIron RX. All ports within each VLAN are untagged. One untagged port within each VLAN is used to connect the BigIron RX for Layer 3 connectivity between the two port-based VLANs.

Figure 9.6 Port-based VLANs 222 and 333



To create the two port-based VLANs shown in Figure 9.6, use the following commands:

```
BigIron RX(config)# vlan 222 by port
BigIron RX(config-vlan-222)# untag e 1/1 to 1/8
BigIron RX(config-vlan-222)# vlan 333 by port
BigIron RX(config-vlan-333)# untag e 1/9 to 1/16
```

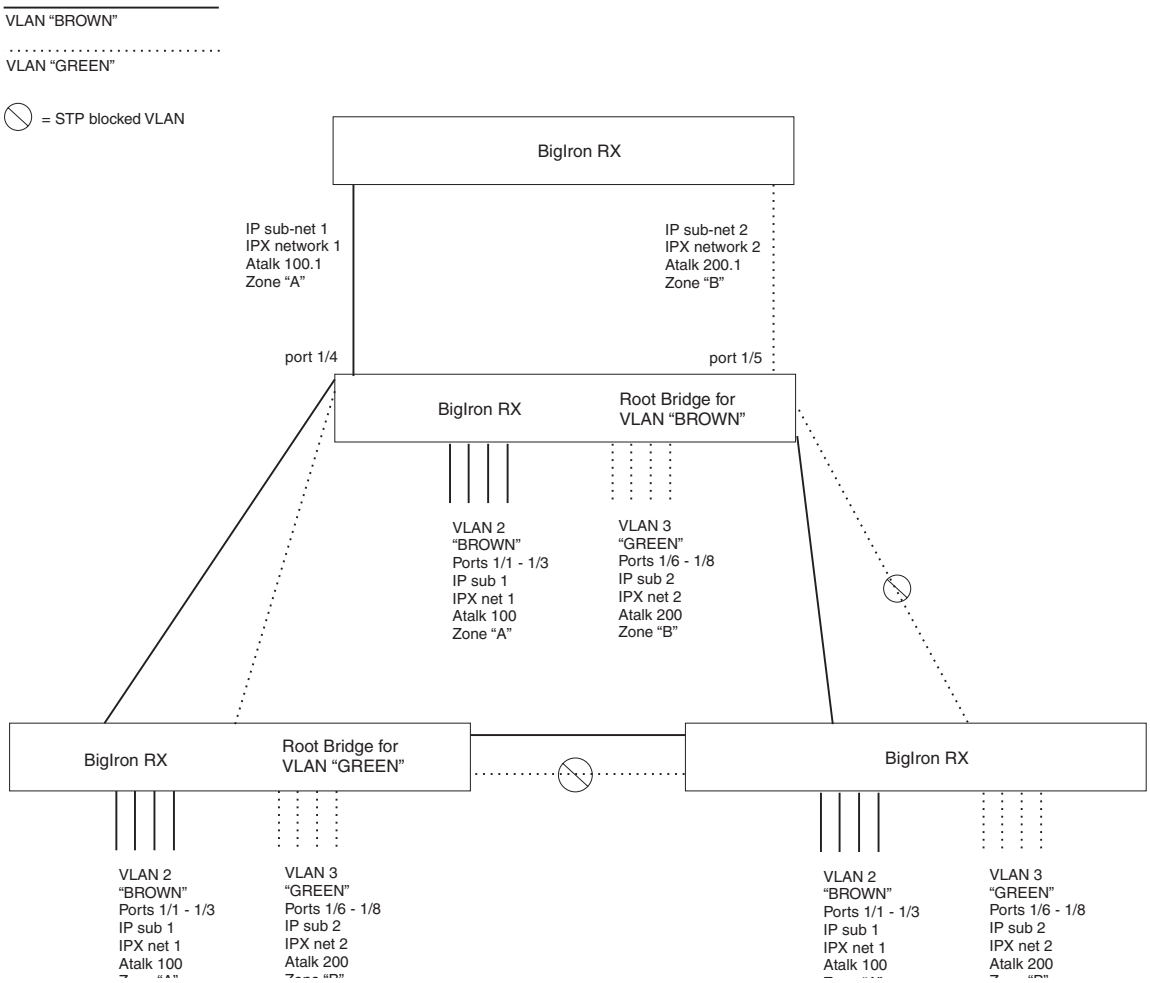
Syntax: vlan <vlan-id> by port

Syntax: untagged ethernet <slot>/<portnum> [to <slot>/<portnum> | ethernet <slot>/<portnum>]

EXAMPLE:

Figure 9.7 shows a more complex port-based VLAN configuration using multiple BigIron RX switches and IEEE 802.1q VLAN tagging. The backbone link connecting the three BigIron RX is tagged. One untagged port within each port-based VLAN BigIron RX A connects each separate network wide Layer 2 broadcast domain to the router for Layer 3 forwarding between broadcast domains. The STP priority is configured to force BigIron RX to be the root bridge for VLANs RED and BLUE. The STP priority on BigIron RX B is configured so that BigIron RX B is the root bridge for VLANs GREEN and BROWN.

Figure 9.7 More complex port-based VLAN



To configure the Port-based VLANs on the BigIron RX, use the following method.

Configuring BigIron RX-A

Enter the following commands to configure BigIron RX-A:

```
BigIron RX> enable
BigIron RX# configure terminal
BigIron RX(config)# hostname BigIron RX-A
BigIron RX-A(config)# vlan 2 name BROWN
BigIron RX-A(config-vlan-2)# untag ethernet 1/1 to 1/4 ethernet 1/17
BigIron RX-A(config-vlan-2)# tag ethernet 1/25 to 1/26
BigIron RX-A(config-vlan-2)# spanning-tree
BigIron RX-A(config-vlan-2)# vlan 3 name GREEN
BigIron RX-A(config-vlan-3)# untag ethernet 1/5 to 1/8 ethernet 1/18
BigIron RX-A(config-vlan-3)# tag ethernet 1/25 to 1/26
BigIron RX-A(config-vlan-3)# spanning-tree
BigIron RX-A(config-vlan-3)# vlan 4 name BLUE
BigIron RX-A(config-vlan-4)# untag ethernet 1/9 to 1/12 ethernet 1/19
BigIron RX-A(config-vlan-4)# tag ethernet 1/25 to 1/26
BigIron RX-A(config-vlan-4)# spanning-tree
BigIron RX-A(config-vlan-4)# spanning-tree priority 500
BigIron RX-A(config-vlan-4)# vlan 5 name RED
BigIron RX-A(config-vlan-5)# untag ethernet 1/13 to 1/16 ethernet 1/20
```



```
BigIron RX-A(config-vlan-5)# tag ethernet 1/25 to 1/26
BigIron RX-A(config-vlan-5)# spanning-tree
BigIron RX-A(config-vlan-5)# spanning-tree priority 500
BigIron RX-A(config-vlan-5)# end
BigIron RX-A# write memory
```

Configuring BigIron RX-B

Enter the following commands to configure BigIron RX-B:

```
BigIron RX> en
BigIron RX# configure terminal
BigIron RX(config)# hostname BigIron RX-B
BigIron RX-B(config)# vlan 2 name BROWN
BigIron RX-B(config-vlan-2)# untag ethernet 1/1 to 1/4
BigIron RX-B(config-vlan-2)# tag ethernet 1/25 to 1/26
BigIron RX-B(config-vlan-2)# spanning-tree
BigIron RX-B(config-vlan-2)# spanning-tree priority 500
BigIron RX-B(config-vlan-2)# vlan 3 name GREEN
BigIron RX-B(config-vlan-3)# untag ethernet 1/5 to 1/8
BigIron RX-B(config-vlan-3)# tag ethernet 1/25 to 1/26
BigIron RX-B(config-vlan-3)# spanning-tree
BigIron RX-B(config-vlan-3)# spanning-tree priority 500
BigIron RX-B(config-vlan-3)# vlan 4 name BLUE
BigIron RX-B(config-vlan-4)# untag ethernet 1/9 to 1/12
BigIron RX-B(config-vlan-4)# tag ethernet 1/25 to 1/26
BigIron RX-B(config-vlan-4)# vlan 5 name RED
BigIron RX-B(config-vlan-5)# untag ethernet 1/13 to 1/16
BigIron RX-B(config-vlan-5)# tag ethernet 1/25 to 1/26
BigIron RX-B(config-vlan-5)# end
BigIron RX-B# write memory
```

Configuring BigIron RX-C

Enter the following commands to configure BigIron RX-C:

```
BigIron RX> en
BigIron RX# configure terminal
BigIron RX(config)# hostname BigIron RX-C
BigIron RX-C(config)# vlan 2 name BROWN
BigIron RX-C(config-vlan-2)# untag ethernet 1/1 to 1/4
BigIron RX-C(config-vlan-2)# tag ethernet 1/25 to 1/26
BigIron RX-C(config-vlan-2)# vlan 3 name GREEN
BigIron RX-C(config-vlan-3)# untag ethernet 1/5 to 1/8
BigIron RX-C(config-vlan-3)# tag ethernet 1/25 to 1/26
BigIron RX-C(config-vlan-3)# vlan 4 name BLUE
BigIron RX-C(config-vlan-4)# untag ethernet 1/9 to 1/12
BigIron RX-C(config-vlan-4)# tag ethernet 1/25 to 1/26
BigIron RX-C(config-vlan-4)# vlan 5 name RED
BigIron RX-C(config-vlan-5)# untag ethernet 1/13 to 1/16
BigIron RX-C(config-vlan-5)# tag ethernet 1/25 to 1/26
BigIron RX-C(config-vlan-5)# end
BigIron RX-C# write memory
```

Syntax: vlan <vlan-id> by port

Syntax: untagged ethernet <slot>/<portnum> [to <slot>/<portnum> | ethernet <slot>/<portnum>]

Syntax: tagged ethernet <slot>/<portnum> [to <slot>/<portnum> | ethernet <slot>/<portnum>]

Syntax: [no] spanning-tree

Syntax: spanning-tree [ethernet <slot>/<portnum> path-cost <value> priority <value>] forward-delay <value> hello-time <value> maximum-age <time> priority <value>

Virtual Routing Interfaces

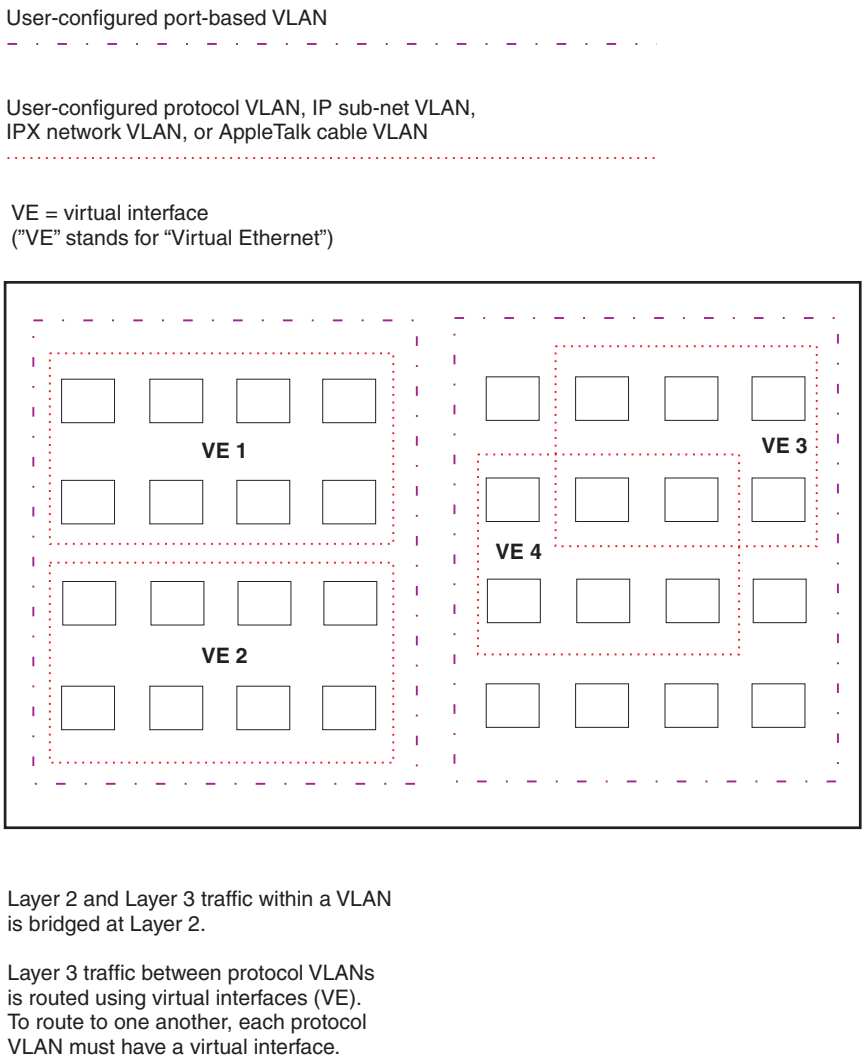
The BigIron RX sends Layer 3 traffic at Layer 2 within a protocol VLAN. However, Layer 3 traffic from one protocol VLAN to another must be routed. If you want the device to be able to send Layer 3 traffic from one protocol VLAN to another, you must configure a virtual routing interface on each protocol VLAN, then configure routing parameters on the virtual routing interfaces.

A virtual routing interface is a logical routing interface that the BigIron RX uses to route Layer 3 protocol traffic between protocol VLANs. It is a logical port on which you can configure Layer 3 routing parameters.

For example, to enable a BigIron RX 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.

Figure 9.8 shows an example of Layer 3 protocol VLANs that use virtual routing interfaces for routing.

Figure 9.8 Use virtual routing interfaces for routing between Layer 3 protocol VLANs



Integrated Switch Routing (ISR)

Foundry Networks' ***Integrated Switch Routing (ISR)*** feature enables VLANs configured on the BigIron RX to route Layer 3 traffic from one protocol VLAN or IP subnet, IPX network, or AppleTalk protocol VLAN to another. Normally, to route traffic from one IP subnet, IPX network, or AppleTalk protocol VLAN to another, you would need to forward the traffic to an external router. The VLANs provide Layer 3 broadcast domains for these protocols but do not in themselves provide routing services for these protocols. This is true even if the source and destination IP subnets, IPX networks, or AppleTalk protocol ranges 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 be able to route from or to. For example, if you configure two IP subnet VLANs on a BigIron RX, you can configure a virtual routing interface on each VLAN, then configure IP routing parameters for the subnets. Thus, the BigIron RX forwards IP subnet broadcasts within each VLAN at Layer 2 but routes Layer 3 traffic between the VLANs using the virtual routing interfaces.

NOTE: The BigIron RX 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 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. The logical interface allows the BigIron RX 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 – 10, you can configure port 5 as a member of an AppleTalk protocol VLAN, an IP protocol VLAN, and an IPX protocol VLAN, and so on.

Routing Between VLANs Using Virtual Routing Interfaces

Routing Between VLANs

The BigIron RX can locally route IP between VLANs defined within a single router. All other routable protocols or protocol VLANs (for example, IPX and AppleTalk) must be routed by another external router capable of routing the protocol.

Virtual Routing Interfaces

You need to configure virtual routing interfaces if an IP, IPX, or AppleTalk protocol VLAN, IP subnet VLAN, AppleTalk protocol VLAN, or IPX network VLAN needs to route protocols to another port-based VLAN on the same router. A virtual routing interface can be associated with the ports in only a single port-based VLAN. Virtual router interfaces must be defined at the highest level of the VLAN hierarchy.

If you do not need to further partition the port-based VLAN by defining separate Layer 3 VLANs, you can define a single virtual routing interface at the port-based VLAN level and enable IP, IPX, and Appletalk routing on a single virtual routing interface.

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, IPX, or Appletalk routing is enabled on a BigIron RX, you can route these protocols 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.

To bridge IP, IPX, or Appletalk at the same time these protocols are being routed, you need to configure an IP protocol, IPX protocol, or Appletalk protocol VLAN and not assign a virtual routing interface to the VLAN. Packets for these protocols are bridged or switched at Layer 2 across ports on the router that are included in the Layer 3 VLAN. If these VLANs are built within port-based VLANs, they can be tagged across a single set of backbone

fibers to create separate Layer 2 switched and Layer 3 routed backbones for the same protocol on a single physical backbone.

Routing Between VLANs Using Virtual Routing Interfaces

Foundry calls the ability to route between VLANs with virtual routing interfaces ***Integrated Switch Routing (ISR)***. There are some important concepts to understand before designing an ISR backbone.

Virtual router interfaces can be defined on port-based, IP protocol, IP subnet, IPX protocol, IPX network, and AppleTalk protocol VLANs.

To create any type of VLAN on a BigIron RX, Layer 2 forwarding must be enabled. When Layer 2 forwarding is enabled, the BigIron RX becomes a Switch on all ports for all non-routable protocols.

If the router interfaces for IP, IPX, or AppleTalk are configured on physical ports, then routing occurs independent of the Spanning Tree Protocol (STP). However, if the router interfaces are defined for any type 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 same protocols 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.

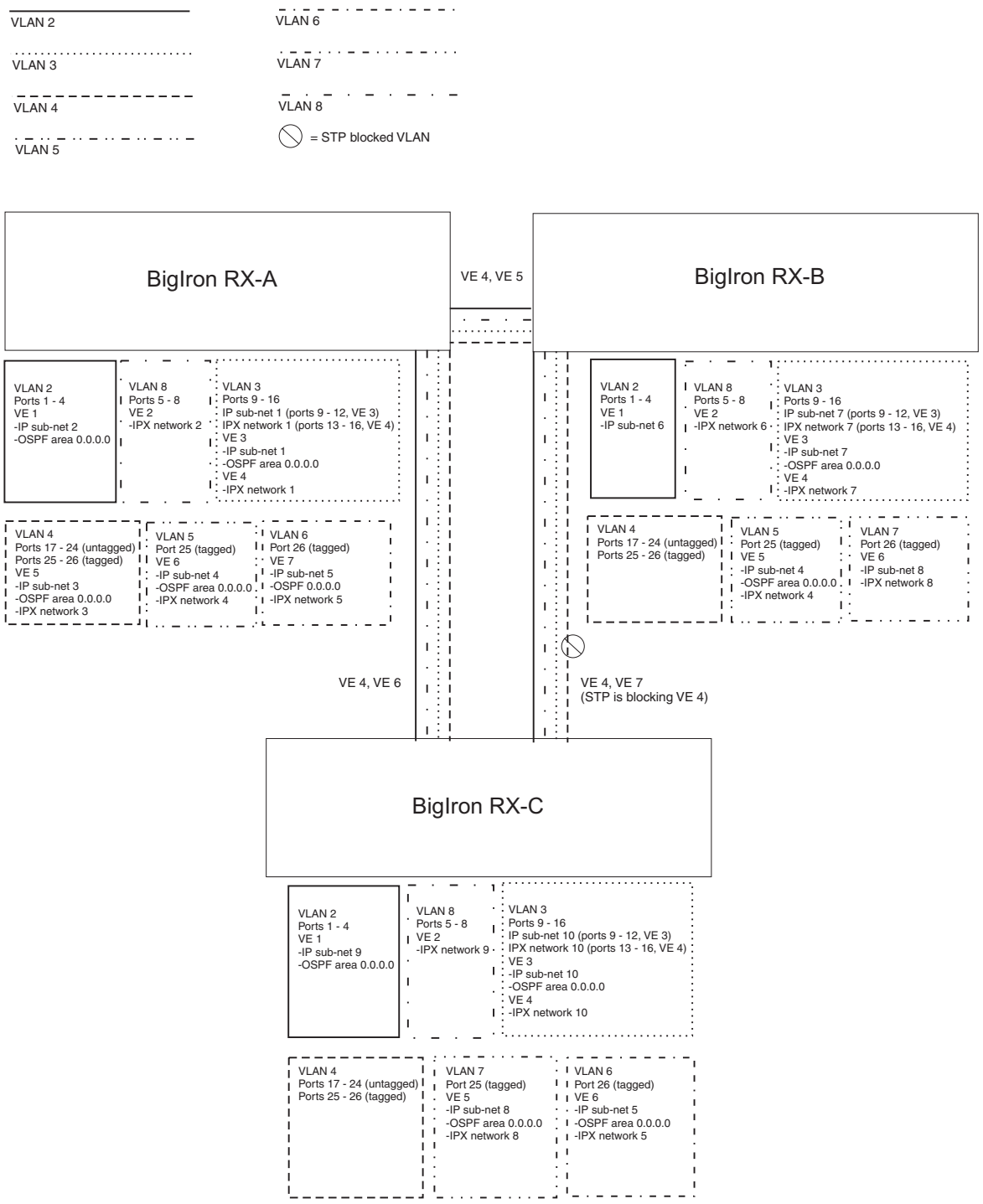
A BigIron RX offers the ability to create a virtual routing interface within a Layer 2 STP port-based VLAN or within each Layer 3 protocol, IP subnet, or IPX network VLAN. This combination of multiple Layer 2 and/or Layer 3 broadcast domains and virtual routing interfaces are the basis for Foundry Networks' very powerful Integrated Switch Routing (ISR) technology. ISR is very flexible and can solve many networking problems. The following example is meant to provide ideas by demonstrating some of the concepts of ISR.

Example: Suppose you want to move routing out to each of three buildings in a network. Remember that the only protocols present on VLAN 2 and VLAN 3 are IP and IPX. Therefore, you can eliminate tagged ports 25 and 26 from both VLAN 2 and VLAN 3 and create new tagged port-based VLANs to support separate IP subnets and IPX networks for each backbone link.

You also need to create unique IP subnets and IPX networks within VLAN 2 and VLAN 3 at each building. This will create a fully routed IP and IPX backbone for VLAN 2 and VLAN 3. However, VLAN 4 has no protocol restrictions across the backbone. In fact there are requirements for NetBIOS and DecNet to be bridged among the three building locations. The IP subnet and IPX network that exists within VLAN 4 must remain a flat Layer 2 switched STP domain. You enable routing for IP and IPX on a virtual routing interface only on BigIron RX-A. This will provide the flat IP and IPX segment with connectivity to the rest of the network. Within VLAN 4 IP and IPX will follow the STP topology. All other IP subnets and IPX networks will be fully routed and have use of all paths at all times during normal operation.

Figure 9.9 shows the configuration described above.

Figure 9.9 Routing between protocol-based VLANs



To configure the Layer 3 VLANs and virtual routing interfaces on the BigIron RX, use the following procedure.

Configuring BigIron RX-A

Enter the following commands to configure BigIron RX-A. The following commands enable OSPF or RIP routing and IPX routing.

```
BigIron RX> en
```



```
No password has been assigned yet...
BigIron RX# configure terminal
BigIron RX(config)# hostname BigIron RX-A
BigIron RX-A(config)# router ospf
BigIron RX-A(config-ospf-router)# area 0.0.0.0 normal
BigIron RX-A(config-ospf-router)# router ipx
ipx routing enabled for next power cycle.
Please save configuration to flash and reboot.
BigIron RX-A(config-ospf-router)#
```

The following commands create the port-based VLAN 2. In the previous example, an external BigIron RX defined the router interfaces for VLAN 2. With ISR, routing for VLAN 2 is done locally within each BigIron RX. Therefore, there are two ways you can solve this problem. One way is to create a unique IP subnet and IPX network VLAN, each with its own virtual routing interface and unique IP or IPX address within VLAN 2 on each BigIron RX. In this example, this is the configuration used for VLAN 3. The second way is to split VLAN 2 into two separate port-based VLANs and create a virtual router interface within each port-based VLAN. Later in this example, this second option is used to create a port-based VLAN 8 to show that there are multiple ways to accomplish the same task with ISR.

You also need to create the Other-Protocol VLAN within port-based VLAN 2 and 8 to prevent unwanted protocols from being Layer 2 switched within port-based VLAN 2 or 8. Note that the only port-based VLAN that requires STP in this example is VLAN 4. You will need to configure the rest of the network to prevent the need to run STP.

```
BigIron RX-A(config-ospf-router)# vlan 2 name IP-Subnet_1.1.2.0/24
BigIron RX-A(config-vlan-2)# untag e 1/1 to 1/4
BigIron RX-A(config-vlan-2)# no spanning-tree
BigIron RX-A(config-vlan-2)# router-interface ve1
BigIron RX-A(config-vlan-2)# other-proto name block_other_protocols
BigIron RX-A(config-vlan-other-proto)# exclude e 1/1 to 1/4
```

Once you have defined the port-based VLAN and created the virtual routing interface, you need to configure the virtual routing interface just as you would configure a physical interface.

```
BigIron RX-A(config-vlan-other-proto)# interface ve1
BigIron RX-A(config-vif-1)# ip address 1.1.2.1/24
BigIron RX-A(config-vif-1)# ip ospf area 0.0.0.0
```

Do the same thing for VLAN 8.

```
BigIron RX-A(config-vif-1)# vlan 8 name IPX_Network2
BigIron RX-A(config-vlan-8)# untag ethernet 1/5 to 1/8
BigIron RX-A(config-vlan-8)# no spanning-tree
BigIron RX-A(config-vlan-8)# router-interface ve 2
BigIron RX-A(config-vlan-8)# other-proto name block-other-protocols
BigIron RX-A(config-vlan-other-proto)# exclude ethernet 1/5 to 1/8
BigIron RX-A(config-vlan-other-proto)# int ve2
BigIron RX-A(config-vif-2)# ipx network 2 ethernet_802.3
BigIron RX-A(config-vif-2)#
```

The next thing you need to do is create VLAN 3. This is very similar to the previous example with the addition of virtual routing interfaces to the IP subnet and IPX network VLANs. Also there is no need to exclude ports from the IP subnet and IPX network VLANs on the router.

```
BigIron RX-A(config-vif-2)# vlan 3 name IP_Sub_&_IPX_Net_VLAN
BigIron RX-A(config-vlan-3)# untag e 1/9 to 1/16
BigIron RX-A(config-vlan-3)# no spanning-tree
BigIron RX-A(config-vlan-3)# ip-subnet 1.1.1.0/24
BigIron RX-A(config-vlan-ip-subnet)# static e 1/9 to 1/12
BigIron RX-A(config-vlan-ip-subnet)# router-interface ve3
BigIron RX-A(config-vlan-ip-subnet)# ipx-network 1 ethernet_802.3
BigIron RX-A(config-vlan-ipx-network)# static e 1/13 to 1/16
BigIron RX-A(config-vlan-ipx-network)# router-interface ve4
BigIron RX-A(config-vlan-ipx-network)# other-proto name block-other-protocols
```



```
BigIron RX-A(config-vlan-other-proto)# exclude e 1/9 to 1/16
BigIron RX-A(config-vlan-other-proto)# interface ve 3
BigIron RX-A(config-vif-3)# ip addr 1.1.1.1/24
BigIron RX-A(config-vif-3)# ip ospf area 0.0.0.0
BigIron RX-A(config-vif-3)# int ve4
BigIron RX-A(config-vif-4)# ipx network 1 ethernet_802.3
BigIron RX-A(config-vif-4)#
```

Now configure VLAN 4. Remember this is a flat segment that, in the previous example, obtained its IP default gateway and IPX router services from an external BigIron RX. In this example, BigIron RX-A will provide the routing services for VLAN 4. You also want to configure the STP priority for VLAN 4 to make BigIron RX-A the root bridge for this VLAN.

```
BigIron RX-A(config-vif-4)# vlan 4 name Bridged_ALL_Protocols
BigIron RX-A(config-vlan-4)# untag ethernet 1/17 to 1/24
BigIron RX-A(config-vlan-4)# tag ethernet 1/25 to 1/26
BigIron RX-A(config-vlan-4)# spanning-tree
BigIron RX-A(config-vlan-4)# spanning-tree priority 500
BigIron RX-A(config-vlan-4)# router-interface ve5
BigIron RX-A(config-vlan-4)# int ve5
BigIron RX-A(config-vif-5)# ip address 1.1.3.1/24
BigIron RX-A(config-vif-5)# ip ospf area 0.0.0.0
BigIron RX-A(config-vif-5)# ipx network 3 ethernet_802.3
BigIron RX-A(config-vif-5)#
```

It is time to configure a separate port-based VLAN for each of the routed backbone ports (Ethernet 25 and 26). If you do not create a separate tagged port-based VLAN for each point-to-point backbone link, you need to include tagged interfaces for Ethernet 25 and 26 within VLANs 2, 3, and 8. This type of configuration makes the entire backbone a single STP domain for each VLAN 2, 3, and 8. In this scenario, the virtual routing interfaces within port-based VLANs 2, 3, and 8 will be accessible using only one path through the network. The path that is blocked by STP is not available to the routing protocols until it is in the STP FORWARDING state.

```
BigIron RX-A(config-vif-5)# vlan 5 name Rtr_BB_to_Bldg.2
BigIron RX-A(config-vlan-5)# tag e 1/25
BigIron RX-A(config-vlan-5)# no spanning-tree
BigIron RX-A(config-vlan-5)# router-interface ve6
BigIron RX-A(config-vlan-5)# vlan 6 name Rtr_BB_to_Bldg.3
BigIron RX-A(config-vlan-6)# tag ethernet 1/26
BigIron RX-A(config-vlan-6)# no spanning-tree
BigIron RX-A(config-vlan-6)# router-interface ve7
BigIron RX-A(config-vlan-6)# int ve6
BigIron RX-A(config-vif-6)# ip addr 1.1.4.1/24
BigIron RX-A(config-vif-6)# ip ospf area 0.0.0.0
BigIron RX-A(config-vif-6)# ipx network 4 ethernet_802.3
BigIron RX-A(config-vif-6)# int ve7
BigIron RX-A(config-vif-7)# ip addr 1.1.5.1/24
BigIron RX-A(config-vif-7)# ip ospf area 0.0.0.0
BigIron RX-A(config-vif-7)# ipx network 5 ethernet_802.3
BigIron RX-A(config-vif-7)#
```

This completes the configuration for BigIron RX-A. The configuration for BigIron RX-B and C is very similar except for a few issues.

- IP subnets and IPX networks configured on BigIron RX-B and BigIron RX-C must be unique across the entire network, except for the backbone port-based VLANs 5, 6, and 7 where the subnet is the same but the IP address must change.
- There is no need to change the default priority of STP within VLAN 4.
- There is no need to include a virtual router interface within VLAN 4.
- The backbone VLAN between BigIron RX-B and BigIron RX-C must be the same at both ends and requires a new VLAN ID. The VLAN ID for this port-based VLAN is VLAN 7.

Configuring BigIron RX-B

Enter the following commands to configure BigIron RX-B.

```
BigIron RX> en
No password has been assigned yet...
BigIron RX# config t
BigIron RX(config)# hostname BigIron RX-B
BigIron RX-B(config)# router ospf
BigIron RX-B(config-ospf-router)# area 0.0.0.0 normal
BigIron RX-B(config-ospf-router)# router ipx
BigIron RX-B(config-ospf-router)# vlan 2 name IP-Subnet_1.1.6.0/24
BigIron RX-B(config-vlan-2)# untag e 1/1 to 1/4
BigIron RX-B(config-vlan-2)# no spanning-tree
BigIron RX-B(config-vlan-2)# router-interface ve1
BigIron RX-B(config-vlan-2)# other-proto name block-other-protocols
BigIron RX-B(config-vlan-other-proto)# exclude e 1/1 to 1/4
BigIron RX-B(config-vlan-other-proto)# int ve1
BigIron RX-B(config-vif-1)# ip addr 1.1.6.1/24
BigIron RX-B(config-vif-1)# ip ospf area 0.0.0.0
BigIron RX-B(config-vif-1)# vlan 8 name IPX_Network6
BigIron RX-B(config-vlan-8)# untag e 1/5 to 1/8
BigIron RX-B(config-vlan-8)# no span
BigIron RX-B(config-vlan-8)# router-int ve2
BigIron RX-B(config-vlan-8)# other-proto name block-other-protocols
BigIron RX-B(config-vlan-other-proto)# exclude e 1/5 to 1/8
BigIron RX-B(config-vlan-other-proto)# int ve2
BigIron RX-B(config-vif-2)# ipx net 6 ethernet_802.3
BigIron RX-B(config-vif-2)# vlan 3 name IP_Sub_&_IPX_Net_VLAN
BigIron RX-B(config-vlan-3)# untag e 1/9 to 1/16
BigIron RX-B(config-vlan-3)# no spanning-tree
BigIron RX-B(config-vlan-3)# ip-subnet 1.1.7.0/24
BigIron RX-B(config-vlan-ip-subnet)# static e 1/9 to 1/12
BigIron RX-B(config-vlan-ip-subnet)# router-interface ve3
BigIron RX-B(config-vlan-ip-subnet)# ipx-network 7 ethernet_802.3
BigIron RX-B(config-vlan-ipx-network)# static e 1/13 to 1/16
BigIron RX-B(config-vlan-ipx-network)# router-interface ve4
BigIron RX-B(config-vlan-ipx-network)# other-proto name block-other-protocols
BigIron RX-B(config-vlan-other-proto)# exclude e 1/9 to 1/16
BigIron RX-B(config-vlan-other-proto)# interface ve 3
BigIron RX-B(config-vif-3)# ip addr 1.1.7.1/24
BigIron RX-B(config-vif-3)# ip ospf area 0.0.0.0
BigIron RX-B(config-vif-3)# int ve4
BigIron RX-B(config-vif-4)# ipx network 7 ethernet_802.3
BigIron RX-B(config-vif-4)# vlan 4 name Bridged_ALL_Protocols
BigIron RX-B(config-vlan-4)# untag ethernet 1/17 to 1/24
BigIron RX-B(config-vlan-4)# tag ethernet 1/25 to 1/26
BigIron RX-B(config-vlan-4)# spanning-tree
BigIron RX-B(config-vlan-4)# vlan 5 name Rtr_BB_to_Bldg.1
BigIron RX-B(config-vlan-5)# tag e 1/25
BigIron RX-B(config-vlan-5)# no spanning-tree
BigIron RX-B(config-vlan-5)# router-interface ve5
BigIron RX-B(config-vlan-5)# vlan 7 name Rtr_BB_to_Bldg.3
BigIron RX-B(config-vlan-7)# tag ethernet 1/26
BigIron RX-B(config-vlan-7)# no spanning-tree
BigIron RX-B(config-vlan-7)# router-interface ve6
BigIron RX-B(config-vlan-7)# int ve5
BigIron RX-B(config-vif-5)# ip addr 1.1.4.2/24
BigIron RX-B(config-vif-5)# ip ospf area 0.0.0.0
```



```
BigIron RX-B(config-vif-5)# ipx network 4 ethernet_802.3
BigIron RX-B(config-vif-5)# int ve6
BigIron RX-B(config-vif-6)# ip addr 1.1.8.1/24
BigIron RX-B(config-vif-6)# ip ospf area 0.0.0.0
BigIron RX-B(config-vif-6)# ipx network 8 ethernet_802.3
BigIron RX-B(config-vif-6)#
```

Configuring BigIron RX-C

Enter the following commands to configure BigIron RX-C.

```
BigIron RX> en
No password has been assigned yet...
BigIron RX# config t
BigIron RX(config)# hostname BigIron RX-C
BigIron RX-C(config)# router ospf
BigIron RX-C(config-ospf-router)# area 0.0.0.0 normal
BigIron RX-C(config-ospf-router)# router ipx
BigIron RX-C(config-ospf-router)# vlan 2 name IP-Subnet_1.1.9.0/24
BigIron RX-C(config-vlan-2)# untag e 1/1 to 1/4
BigIron RX-C(config-vlan-2)# no spanning-tree
BigIron RX-C(config-vlan-2)# router-interface ve1
BigIron RX-C(config-vlan-2)# other-proto name block-other-protocols
BigIron RX-C(config-vlan-other-proto)# exclude e 1/1 to 1/4
BigIron RX-C(config-vlan-other-proto)# int ve1
BigIron RX-C(config-vif-1)# ip addr 1.1.9.1/24
BigIron RX-C(config-vif-1)# ip ospf area 0.0.0.0
BigIron RX-C(config-vif-1)# vlan 8 name IPX_Network9
BigIron RX-C(config-vlan-8)# untag e 1/5 to 1/8
BigIron RX-C(config-vlan-8)# no span
BigIron RX-C(config-vlan-8)# router-int ve2
BigIron RX-C(config-vlan-8)# other-proto name block-other-protocols
BigIron RX-C(config-vlan-other-proto)# exclude e 1/5 to 1/8
BigIron RX-C(config-vlan-other-proto)# int ve2
BigIron RX-C(config-vif-2)# ipx net 9 ethernet_802.3
BigIron RX-C(config-vif-2)# vlan 3 name IP_Sub_&_IPX_Net_VLAN
BigIron RX-C(config-vlan-3)# untag e 1/9 to 1/16
BigIron RX-C(config-vlan-3)# no spanning-tree
BigIron RX-C(config-vlan-3)# ip-subnet 1.1.10.0/24
BigIron RX-C(config-vlan-ip-subnet)# static e 1/9 to 1/12
BigIron RX-C(config-vlan-ip-subnet)# router-interface ve3
BigIron RX-C(config-vlan-ip-subnet)# ipx-network 10 ethernet_802.3
BigIron RX-C(config-vlan-ipx-network)# static e 1/13 to 1/16
BigIron RX-C(config-vlan-ipx-network)# router-interface ve4
BigIron RX-C(config-vlan-ipx-network)# other-proto name block-other-protocols
BigIron RX-C(config-vlan-other-proto)# exclude e 1/9 to 1/16
BigIron RX-C(config-vlan-other-proto)# interface ve 3
BigIron RX-C(config-vif-3)# ip addr 1.1.10.1/24
BigIron RX-C(config-vif-3)# ip ospf area 0.0.0.0
BigIron RX-C(config-vif-3)# int ve4
BigIron RX-C(config-vif-4)# ipx network 10 ethernet_802.3
BigIron RX-C(config-vif-4)# vlan 4 name Bridged_ALL_Protocols
BigIron RX-C(config-vlan-4)# untag ethernet 1/17 to 1/24
BigIron RX-C(config-vlan-4)# tag ethernet 1/25 to 1/26
BigIron RX-C(config-vlan-4)# spanning-tree
BigIron RX-C(config-vlan-4)# vlan 7 name Rtr_BB_to_Bldg.2
BigIron RX-C(config-vlan-7)# tag e 1/25
BigIron RX-C(config-vlan-7)# no spanning-tree
BigIron RX-C(config-vlan-7)# router-interface ve5
BigIron RX-C(config-vlan-7)# vlan 6 name Rtr_BB_to_Bldg.1
```



```
BigIron RX-C(config-vlan-6)# tag ethernet 1/26
BigIron RX-C(config-vlan-6)# no spanning-tree
BigIron RX-C(config-vlan-6)# router-interface ve6
BigIron RX-C(config-vlan-6)# int ve5
BigIron RX-C(config-vif-5)# ip addr 1.1.8.2/24
BigIron RX-C(config-vif-5)# ip ospf area 0.0.0.0
BigIron RX-C(config-vif-5)# ipx network 8 ethernet_802.3
BigIron RX-C(config-vif-5)# int ve6
BigIron RX-C(config-vif-6)# ip addr 1.1.5.2/24
BigIron RX-C(config-vif-6)# ip ospf area 0.0.0.0
BigIron RX-C(config-vif-6)# ipx network 5 ethernet_802.3
BigIron RX-C(config-vif-6)#
```

Configuring VLAN Groups

To simplify 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, see "Allocating Memory for More VLANs or Virtual Routing Interfaces" on page 9-27.

Configuring a VLAN Group

To configure a VLAN group, enter commands such as the following:

```
BigIron RX(config)# vlan-group 1 vlan 2 to 1000
BigIron RX(config-vlan-group-1)# tagged e 1/1 to 1/2
```

The first command begins configuration for VLAN group 1, creates VLANs 2 through 1000 and assigns them to the group. The second command adds ports 1/1 and 1/2 as tagged ports to the group. Since all the VLANs in the group share the ports, you must add the ports as tagged ports.

Syntax: vlan-group <num> vlan <vlan-id> to <vlan-id>

Syntax: tagged ethernet [to <slot/port> | ethernet <slot/port>]

The <num> parameter with the **vlan-group** command specifies the VLAN group ID and can be from 1 – 32.

The **vlan** <vlan-id> **to** <vlan-id> parameters specify a range (with no gaps) of VLAN IDs. Specify the low VLAN ID first and the high VLAN ID second. The command adds all the specified VLANs to the VLAN 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. See "Allocating Memory for More VLANs or Virtual Routing Interfaces" on page 9-27.

If a VLAN within the range you specify is already configured, the CLI does not add the group but instead displays an error message. In this case, create the group by specifying a valid contiguous range. Then add more VLANs to the group after the CLI changes to the configuration level for the group. See the following example.

You can add and remove individual VLANs or VLAN ranges from at 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>]

Displaying Information about VLAN Groups

To display VLAN group configuration information, enter the following command:

```
BigIron RX# show vlan-group
vlan-group 1 vlan 2 to 20
 tagged ethe 1/1 to 1/2
!
vlan-group 2 vlan 21 to 40
 tagged ethe 1/1 to 1/2
!
```

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.

Displaying the VLAN Group

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
 router-interface-group

If you have enabled display of subnet masks in CIDR notation, the IP address
information is shown as follows:  10.10.10.1/24.
```

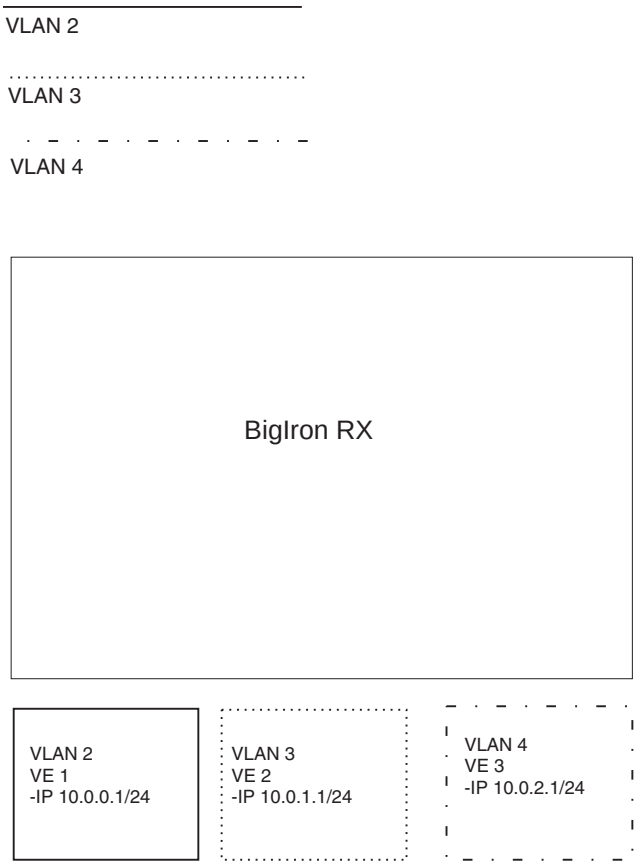
Configuring the Same IP Subnet Address on Multiple Port-Based VLANs

For a BigIron RX to route between port-based VLANs, you must add a virtual routing interface to each VLAN. Generally, you also configure a unique IP subnet address on each virtual routing interface. For example, if you have three port-based VLANs, you add a virtual routing interface to each VLAN, then add a separate IP subnet address to each virtual routing interface. The IP address on each of the virtual routing interfaces must be in a separate subnet. The BigIron RX routes Layer 3 traffic between the subnets using the subnet addresses.

NOTE: Before using the method described in this section, see “Configuring VLAN Groups” on page 9-23. You might be able to achieve the results you want using the methods in that section instead.

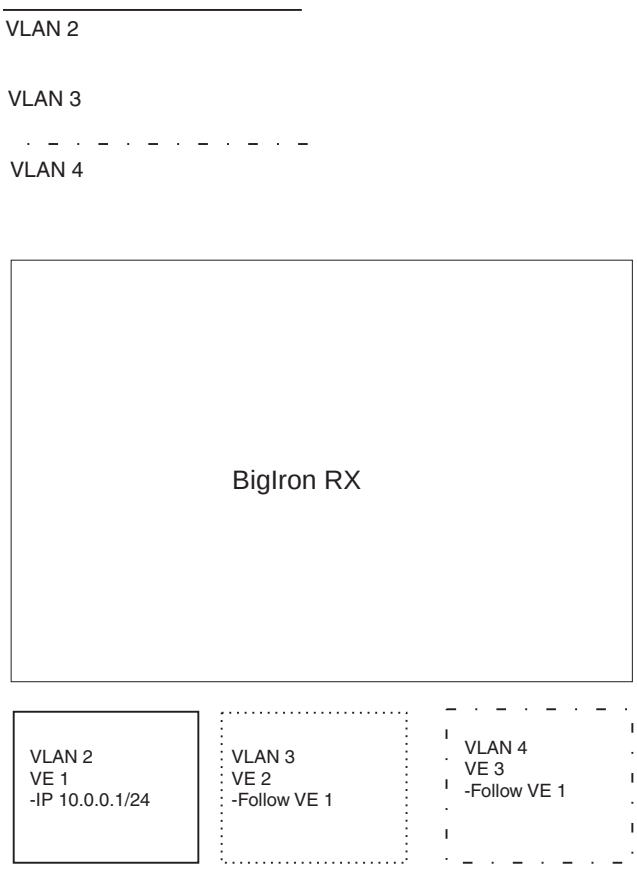
Figure 9.10 shows an example of this type of configuration.

Figure 9.10 Multiple port-based VLANs with separate protocol addresses



As shown in this example, each VLAN has a separate IP subnet address. If you need to conserve IP subnet addresses, you can configure multiple VLANs with the same IP subnet address, as shown in Figure 9.11.

Figure 9.11 Multiple port-based VLANs with the same protocol address



Each VLAN still requires a separate virtual routing interface. However, all three VLANs now use the same IP subnet address.

In addition to conserving IP subnet addresses, this feature allows containment of Layer 2 broadcasts to segments within an IP subnet. For ISP environments where the same IP subnet is allocated to different customers, placing each customer in a separate VLAN allows all customers to share the IP subnet address, while at the same time isolating them from one another's Layer 2 broadcasts.

NOTE: You can provide redundancy to an IP subnet address that contains multiple VLANs using a pair of BigIron RX switches configured for Foundry's VRRP (Virtual Router Redundancy Protocol).

The BigIron RX performs proxy Address Resolution Protocol (ARP) for hosts that want to send IP traffic to hosts in other VLANs that are sharing the same IP subnet address. If the source and destination hosts are in the same VLAN, the BigIron RX does not need to use ARP.

- If a host attached to one VLAN sends an ARP message for the MAC address of a host in one of the other VLANs using the same IP subnet address, the BigIron RX performs a proxy ARP on behalf of the other host. The BigIron RX then replies to the ARP by sending the virtual routing interface MAC address. The BigIron RX uses the same MAC address for all virtual routing interfaces.

  When the host that sent the ARP then sends a unicast packet addressed to the virtual routing interface's MAC address, the device switches the packet on Layer 3 to the destination host on the VLAN.

NOTE: If the BigIron RX's ARP table does not contain the requested host, the BigIron RX forwards the ARP request on Layer 2 to the same VLAN as the one that received the ARP request. Then the device sends an ARP for the destination to the other VLANs that are using the same IP subnet address.

- If the destination is in the same VLAN as the source, the BigIron RX does not need to perform a proxy ARP.

To configure multiple VLANs to use the same IP subnet address:

- Configure each VLAN, including adding tagged or untagged ports.
- Configure a separate virtual routing interface for each VLAN, but do not add an IP subnet address to more than one of the virtual routing interfaces.
- Configure the virtual routing interfaces that do not have the IP subnet address to "follow" the virtual routing interface that does have the address.

To configure the VLANs shown in Figure 9.11, you could enter the following commands.

```
BigIron RX(config)# vlan 1 by port
BigIron RX(config-vlan-1)# untag ethernet 1/1
BigIron RX(config-vlan-1)# tag ethernet 1/8
BigIron RX(config-vlan-1)# router-interface ve 1
```

The commands above configure port-based VLAN 1. The VLAN has one untagged port (1/1) and a tagged port (1/8). In this example, all three VLANs contain port 1/8 so the port must be tagged to allow the port to be in multiple VLANs. You can configure VLANs to share a Layer 3 protocol interface regardless of tagging. A combination of tagged and untagged ports is shown in this example to demonstrate that sharing the interface does not change other VLAN features.

Notice that each VLAN still requires a unique virtual routing interface.

The following commands configure port-based VLANs 2 and 3.

```
BigIron RX(config-vlan-1)# vlan 2 by port
BigIron RX(config-vlan-2)# untag ethernet 1/2
BigIron RX(config-vlan-2)# tag ethernet 1/8
BigIron RX(config-vlan-2)# router-interface ve 2
BigIron RX(config-vlan-2)# vlan 3 by port
BigIron RX(config-vlan-3)# untag ethernet 1/5 to 1/6
BigIron RX(config-vlan-3)# tag ethernet 1/8
BigIron RX(config-vlan-3)# router-interface ve 3
```

The following commands configure an IP subnet address on virtual routing interface 1.

```
BigIron RX(config-vlan-3)# interface ve 1
BigIron RX(config-vif-1)# ip address 10.0.0.1/24
```

Allocating Memory for More VLANs or Virtual Routing Interfaces

By default, you can configure up to 512 VLANs and virtual routing interfaces on the BigIron RX. This is the default maximum. However, BigIron RX can support up to 4095 VLANs and 4095 virtual routing interfaces, but VLAN IDs 0, 4092 and 4095 are reserved; therefore, only 4093 VLANs are user configurable.

NOTE: If many of your VLANs will have an identical configuration, you might want to configure VLAN groups.

To increase the maximum number of VLANs you can configure, 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: system-max vlan <num>

The <num> parameter specifies the maximum number of VLANs. Enter 1 – 4093 since VLAN IDs 0, 4092 and 4095 are reserved.

Configuring Super Aggregated VLANs

You can aggregate multiple VLANs within another VLAN. This feature allows you to construct Layer 2 paths and channels. This feature is particularly useful for Virtual Private Network (VPN) applications in which you need to provide a private, dedicated Ethernet connection for an individual client to transparently reach its subnet across multiple networks.

Conceptually, the paths and channels are similar to Asynchronous Transfer Mode (ATM) 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.

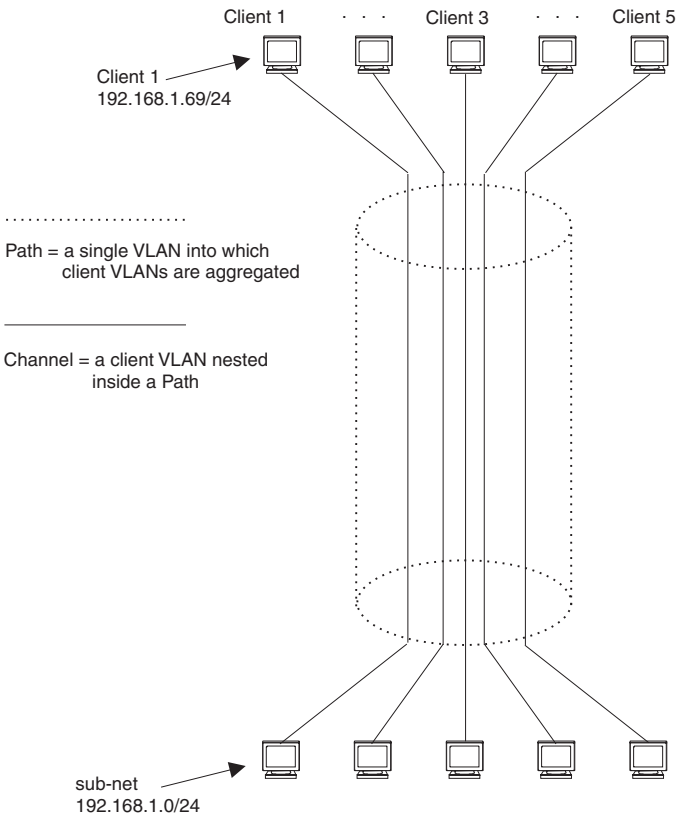
You can aggregate up to 4094 VLANs within another VLAN. This provides a total VLAN capacity on one BigIron RX of 16,760,836 channels (4094 * 4094).

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.

The feature allows point-to-point and point-to-multipoint connections.

Figure 9.12 shows a conceptual picture of the service that aggregated VLANs provide. Aggregated VLANs provide a path for multiple client channels. The channels do not receive traffic from other channels. Thus, each channel is a private link.

Figure 9.12 Conceptual Model of the Super Aggregated VLAN Application

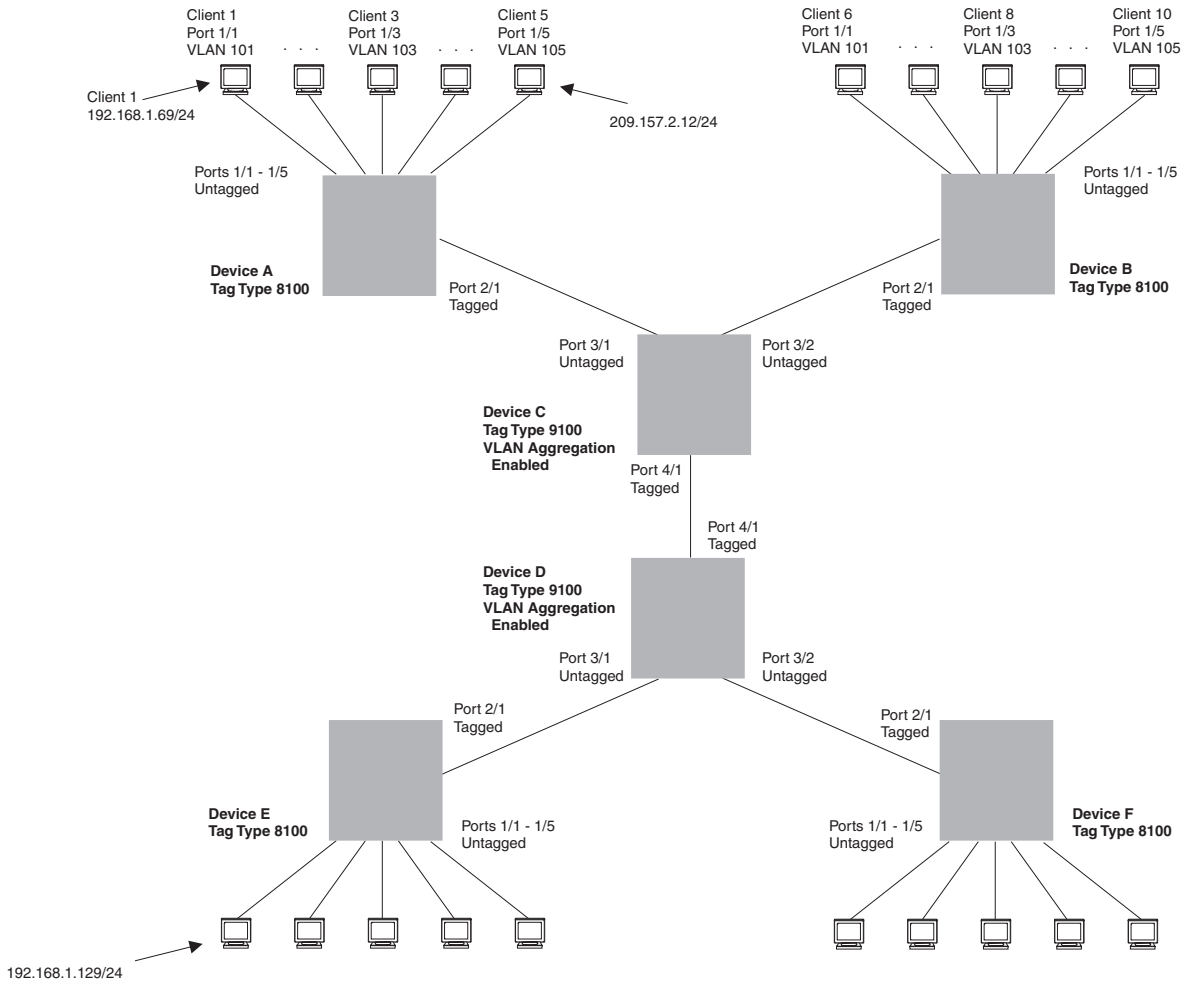


Each client connected to the edge device is in its own port-based VLAN, which is like an ATM channel. All the clients' VLANs are aggregated by the edge device into a single VLAN for connection to the core. The single VLAN that aggregates the clients' VLANs is like an ATM path.

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 9.13 shows an example application that uses aggregated VLANs. This configuration includes the client connections shown in Figure 9.12.

Figure 9.13 Example Super Aggregated VLAN Application



In this example, 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, perform the following tasks:

- 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.
- 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.

Configuring Aggregated VLANs on an Edge Device

To configure the aggregated VLANs on device A in Figure 9.13 on page 9-29, enter the following commands:

```
BigIron RX(config)# vlan 101 by port
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 by port
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 by port
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 by port
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 by port
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> [by port]

Syntax: [no] untagged | tagged ethernet <slot/port> [to <slot/port> | ethernet <slot/port>]

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.

Configuring Aggregated VLANs on a Core Device

To configure the aggregated VLANs on device C in Figure 9.13 on page 9-29, enter the following commands:

```
BigIron RX(config)# tag-type 9100
BigIron RX(config)# aggregated-vlan
BigIron RX(config)# vlan 101 by port
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 by port
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 9.13 on page 9-29.

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 9.13 on page 9-29 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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
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 by port
```



```
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 9.13 on page 9-29 is symmetrical, the commands for configuring device E are identical to the commands for configuring device A.

```
BigIron RX-E(config)# vlan 101 by port
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 by port
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 by port
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 by port
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 by port
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 9.13 on page 9-29 are symmetrical, the configuration of device F is also identical to the configuration of device A and device B.

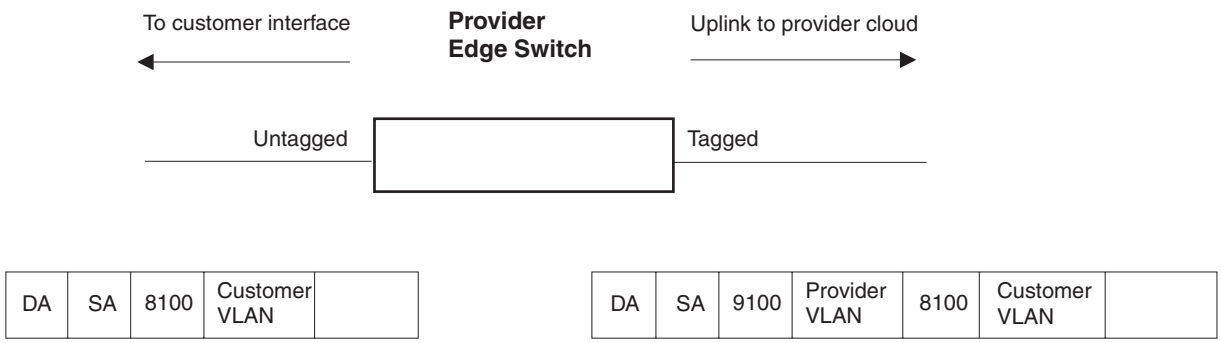
```
BigIron RX-F(config)# vlan 101 by port
BigIron RX-F(config-vlan-101)# tagged ethernet 2/1
BigIron RX-F(config-vlan-101)# untagged ethernet 1/1
BigIron RX-F(config-vlan-101)# exit
BigIron RX-F(config)# vlan 102 by port
BigIron RX-F(config-vlan-102)# tagged ethernet 2/1
BigIron RX-F(config-vlan-102)# untagged ethernet 1/2
BigIron RX-F(config-vlan-102)# exit
BigIron RX-F(config)# vlan 103 by port
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 by port
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 by port
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

You can configure 802.1Q tag-types on a group of ports, thereby enabling the creation of two identical 802.1Q tags (802.1Q-in-Q tagging) on a single device. This improves SAV interoperability between Foundry devices and other vendors' devices that support the 802.1Q tag-types, but is not very flexible with the tag-types they accept.

Figure 9.14 shows an 802.1Q configuration example.

Figure 9.14 802.1Q Configuration Example



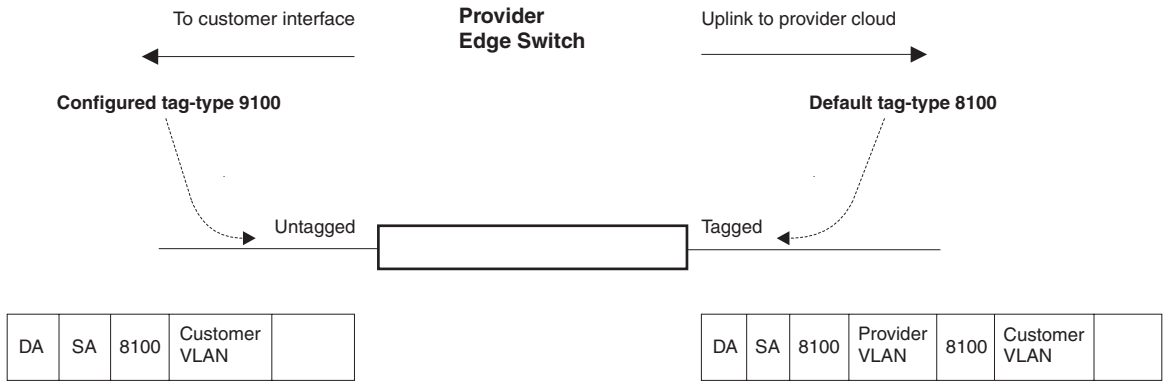
As shown in Figure 9.14, the ports to customer interfaces are untagged, whereas the uplink ports to the provider cloud are tagged, because multiple client VLANs share the uplink to the provider cloud. In this example, the BigIron RX treats the customer's private VLAN ID and 8100 tag type as normal payload, 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 are BigIron RX that can use the 9100 tag type, the data gets switched along the network. However, devices along the provider's cloud that do not support the 9100 tag type may not properly handle the packets.

802.1Q-in-Q tagging enables you to configure 802.1Q tag-types on a group of ports, thereby enabling the creation of two identical 802.1Q tags (802.1Q-in-Q tagging) on a single device. This enhancement improves SAV interoperability between Foundry devices and other vendors' devices that support the 802.1Q tag-types, but are not very flexible with the tag-types they accept.

Figure 9.17 shows an example application of the 802.1Q-in-Q enhancement.

Figure 9.15 802.1Q-in-Q Configuration Example



In Figure 9.17, 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 BigIron RX 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

- 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 BigIron RX 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 BigIron RX automatically removes the 802.1Q tag-type from all ports within the same port region.
- The BigIron RX 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 on the untagged edge links (the customer ports) to any value other than the 802.1Q tag for incoming traffic. For example, in Figure 9.16, 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 9.16, enter commands such as the following on the untagged edge links of devices C and D:

```
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 actually applies to ports 9 – 16.

Syntax: [no] tag-type <num> [ethernet <slot/port> [to <slot/port>]]

The **<num>** parameter specifies the tag-type number and can be a hexadecimal value from 0 - ffff. The default is 8100.

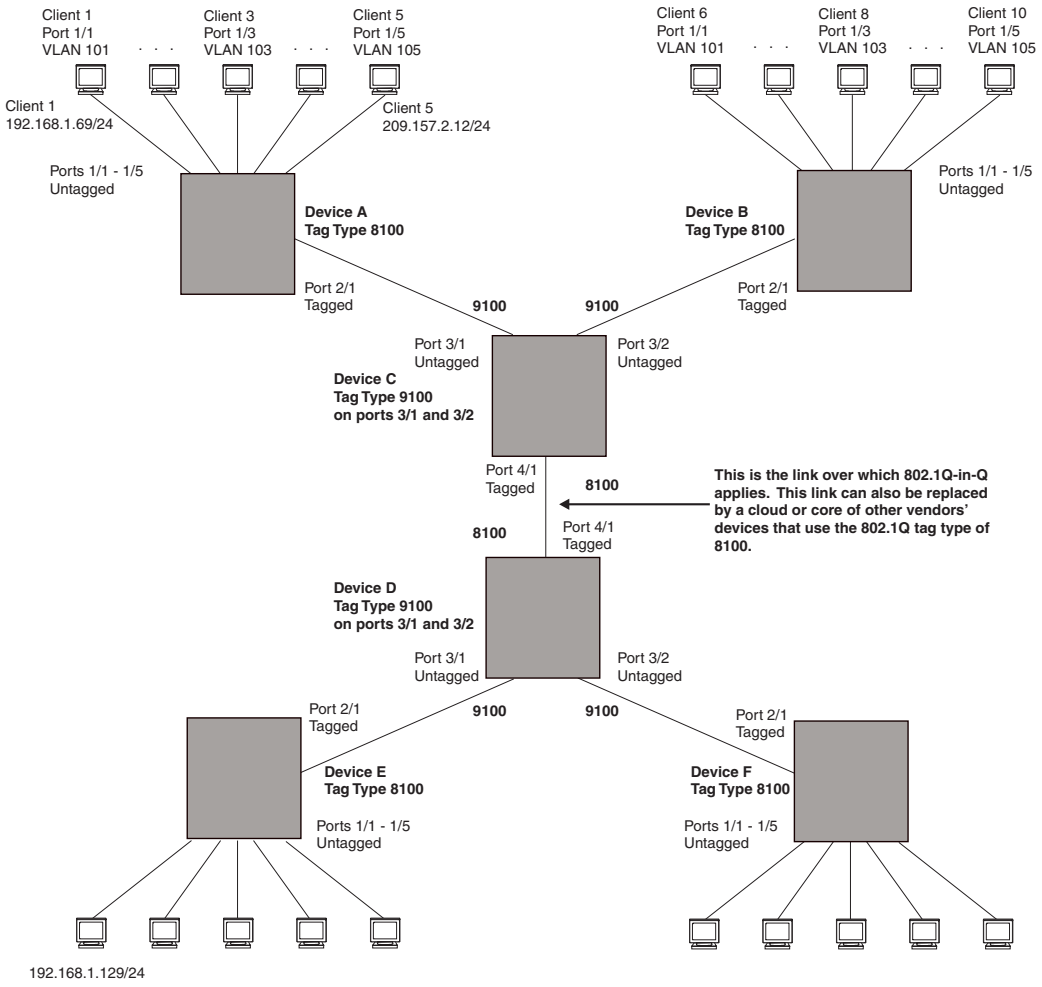
The **e <port number> to <port number>** parameter specifies the port(s) that will use the defined 802.1Q tag. This parameter operates with the following rules:

- If you specify a single port number, the 802.1Q tag applies to all ports within the port region. For example, if you enter the command **tag-type 9100 e 1**, the BigIron RX automatically applies the 802.1Q tag to ports 1 – 8 since all of these ports are in the same port region. 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 9.16 shows an example 802.1Q-in-Q configuration.

Figure 9.16 Example 802.1Q-in-Q Configuration



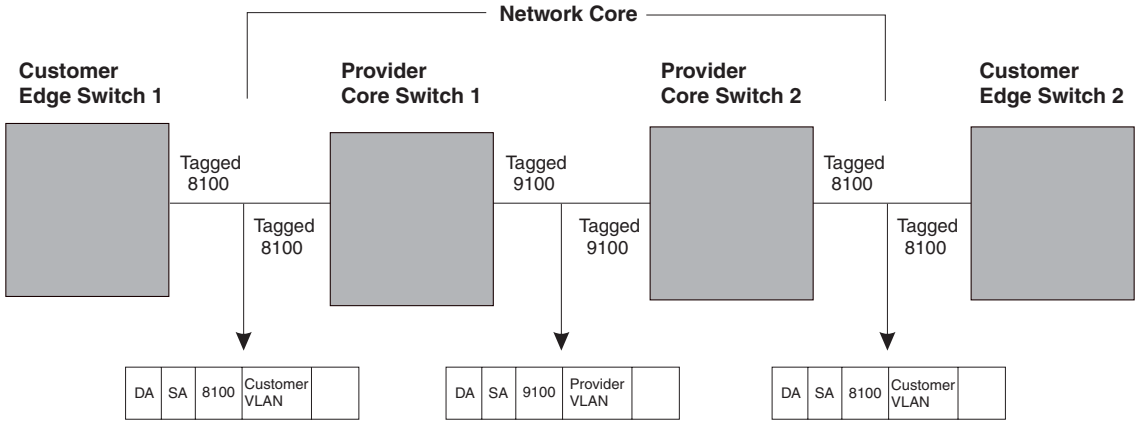
Configuring 802.1q Tag-type Translation

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 9.17 shows a basic example application of the 802.1q tag-type translation feature.

Figure 9.17 802.1q Tag-type Translation Configuration Example 1



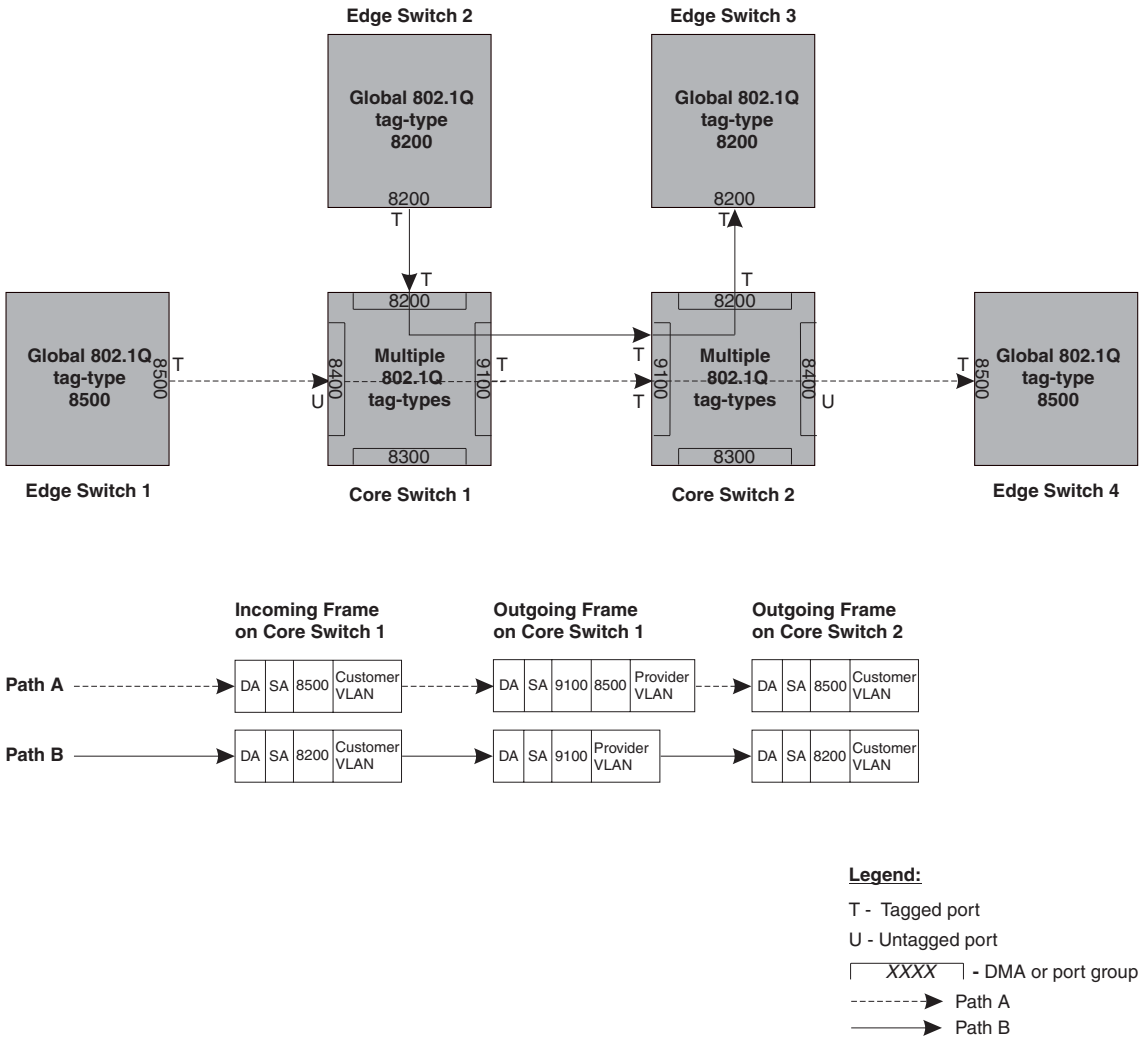
As illustrated in Figure 9.17, the devices process the packet as follows:

- 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 9.17 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 9.18 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 9.18 802.1q Tag-type Translation Configuration Example 2



As illustrated in Figure 9.18, 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, see “Configuring 802.1q Tag-type Translation” on page 9-36.

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 BigIron RX 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 BigIron RX automatically removes the 802.1q tag-type from all ports within the same port region.
- Foundry 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 9.1 describes the granularity at which each of the BigIron RX interface modules can have 802.1Q tag-types assigned.

Table 9.1: 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 (see Figure 9.17). 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/port> [to <slot/port>]]

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/port> [**to** <slot/port>] parameter specifies the port(s) that will use the defined 802.1q tag-type. This parameter operates with the following rules:

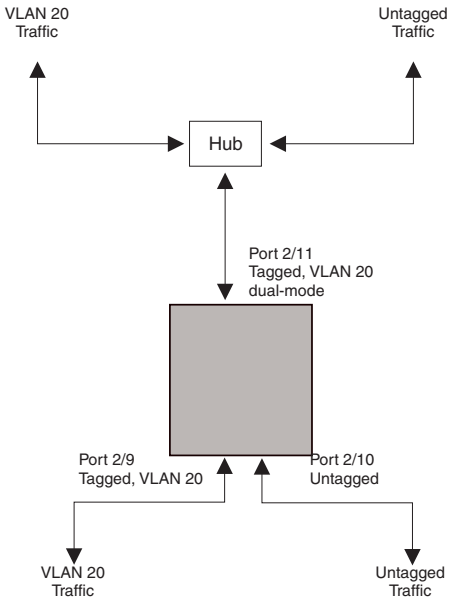
- If you specify a single port number, the 802.1q tag-type applies to all ports within the port region. For example, if you enter the command **tag-type 9100 e 1**, the BigIron RX automatically applies the 802.1q tag to ports 1 – 8 since all of these ports are in the same port region (controlled by the 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.

Dual-Mode VLAN Ports

Configuring a tagged port as a dual-mode port allows it to accept and transmit both tagged traffic and untagged traffic at the same time. A dual-mode port accepts and transmits frames belonging to VLANs configured for the port, as well as frames belonging to the default VLAN (that is, untagged traffic).

For example, in Figure 9.19, port 2/11 is a dual-mode port belonging to VLAN 20. Traffic for VLAN 20, as well as traffic for the default VLAN, flows from a hubs to this port. The dual-mode feature allows traffic for VLAN 20 and untagged traffic to go through the port at the same time.

Figure 9.19 Dual-mode VLAN port example



To enable the dual-mode feature on port 2/11 in Figure 9.19:

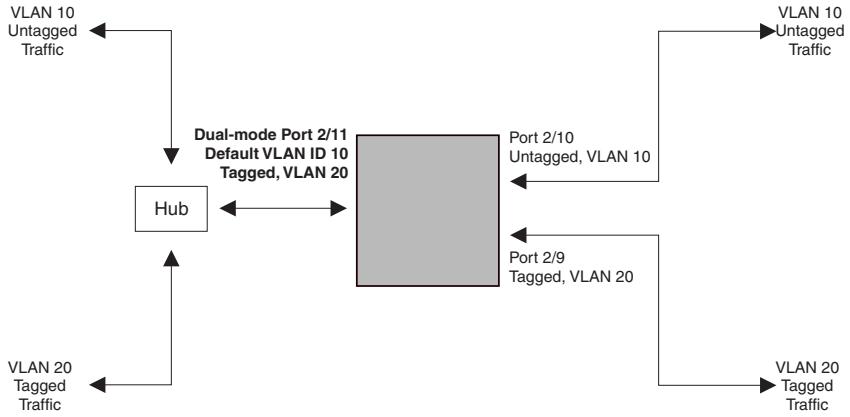
```
BigIron RX(config)# vlan 20
BigIron RX(config-vlan-20)# tagged e 2/11
BigIron RX(config-vlan-20)# tagged e 2/9
BigIron RX(config-vlan-20)# int e 2/11
BigIron RX(config-if-e100-2/11)# dual-mode
BigIron RX(config-if-e100-2/11)# exit
```

Syntax: [no] dual-mode

A dual-mode port accepts and transmits frames belonging to VLANs configured for the port, as well as frames belonging to the DEFAULT-VLAN (VLAN 1). Traffic for the DEFAULT-VLAN is transmitted untagged, and traffic for other VLANs is tagged.

You can configure a dual-mode port to transmit traffic for a specified VLAN (other than the DEFAULT-VLAN) as untagged, while transmitting traffic for other VLANs as tagged. Figure 9.20 illustrates this enhancement.

Figure 9.20 Specifying a default VLAN ID for a dual-mode port



In Figure 9.20, tagged port 2/11 is a dual-mode port belonging to VLANs 10 and 20. The default VLAN assigned to this dual-mode port is 10. This means that the port transmits tagged traffic on VLAN 20 (and all other VLANs to which the port belongs) and transmits untagged traffic on VLAN 10.

The dual-mode feature allows tagged traffic for VLAN 20 and untagged traffic for VLAN 10 to go through port 2/11 at the same time. A dual-mode port transmits only untagged traffic on its default VLAN (that is, either VLAN 1, or a user-specified VLAN ID), and only tagged traffic on all other VLANs.

The following commands configure VLANs 10 and 20 in Figure 9.20. Tagged port 2/11 is added to VLANs 10 and 20, then designated a dual-mode port whose specified default VLAN is 10. In this configuration, port 2/11 transmits only untagged traffic on VLAN 10 and only tagged traffic on VLAN 20.

```
BigIron RX(config)# vlan 10 by port
BigIron RX(config-vlan-10)# untagged e 2/10
BigIron RX(config-vlan-10)# tagged e 2/11
BigIron RX(config-vlan-10)# exit

BigIron RX(config)# vlan 20 by port
BigIron RX(config-vlan-20)# tagged e 2/9
BigIron RX(config-vlan-20)# tagged e 2/11
BigIron RX(config-vlan-20)# exit

BigIron RX(config)# int e 2/11
BigIron RX(config-if-e100-2/11)# dual-mode 10
BigIron RX(config-if-e100-2/11)# exit
```

Syntax: [no] dual-mode [<vlan-id>]

Notes:

- If you do not specify a <vlan-id> in the dual mode command, the port's default VLAN is set to 1. The port transmits untagged traffic on the DEFAULT-VLAN.
- The dual-mode feature is disabled by default. Only tagged ports can be configured as dual-mode ports.
- In trunk group, either all of the ports must be dual-mode, or none of them can be.

The **show vlan** command displays a separate row for dual-mode ports on each VLAN. For example:

```
BigIron RX(config)# show vlan
Total PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 16

legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree Off
 Untagged Ports: (S1)  1  2  3  4  5  6  7  8
 Untagged Ports: (S2)  1  2  3  4  5  6  7  8 12 13 14 15 16 17 18 19
 Untagged Ports: (S2) 20 21 22 23 24
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: None
PORT-VLAN 10, Name [None], Priority level0, Spanning tree Off
 Untagged Ports: (S2) 10
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: (S2) 11
PORT-VLAN 20, Name [None], Priority level0, Spanning tree Off
 Untagged Ports: None
   Tagged Ports: (S2)  9
   Uplink Ports: None
 DualMode Ports: (S2) 11
```

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 BigIron RX 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 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
BigIron RX(config-vlan-2)# exit
BigIron RX(config)# reload
```

Syntax: [no] multicast-flooding

After entering the **multicast-flooding** command for a VLAN, you must reboot the BigIron RX to activate the feature.

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.

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 BigIron RX performs *hardware* flooding for unknown unicast packets. However, if uplink VLANs or protocol 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

Displaying VLAN Information

After you configure the VLANs, you can view and verify the configuration.

Displaying System-Wide VLAN Information

Enter the following command at any CLI level:

```
BigIron RX(config)# show vlan

Total PORT-VLAN entries: 2
Maximum PORT-VLAN entries: 8
legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree Off
 Untagged Ports: (S2)  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
 Untagged Ports: (S2) 17 18 19 20 21 22 23 24
 Untagged Ports: (S4)  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
 Untagged Ports: (S4) 17 18 19 20 21 22 23 24
   Tagged Ports: None

PORT-VLAN 10, Name IP_VLAN, Priority level0, Spanning tree Off
 Untagged Ports: (S1)  1  2  3  4  5  6
   Tagged Ports: None
```


Displaying VLAN Information for Specific Ports

To display VLAN information for all the VLANs of which port 7/1 is a member, enter the following command:

```
BigIron RX(config)# show vlan e 7/1
Total PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 8

legend: [S=Slot]

PORT-VLAN 100, Name [None], Priority level0, Spanning tree Off
 Untagged Ports: (S7)  1  2  3  4
   Tagged Ports: None

 IP-subnet VLAN 207.95.11.0 255.255.255.0,
 Static ports: (S7)  1  2
 Exclude ports: None
```

Syntax: show vlan [<vlan-id> | ethernet <slot/port> | detail | | begin <expression> | exclude <expression> | include <expression>]

The <vlan-id> parameter specifies a VLAN for which you want to display the configuration information.

The **ethernet** <slot/port> parameter specifies a port. The command lists all the VLAN memberships for the port.

Chapter 10
Configuring Spanning Tree Protocol

This chapter the Spanning Tree Protocol (STP) that are supported in the BigIron RX. The chapter contains the following sections:

IEEE 802.1D Spanning Tree Protocol (STP)

The BigIron RX supports Spanning Tree Protocol (STP) as described in the IEEE 802.10-1998 specification. STP eliminates Layer 2 loops in networks, by selectively blocking some ports and allowing other ports to forward traffic, based on configurable bridge and port parameters. STP also ensures that the least cost path is taken when multiple paths exist between ports or VLANs. If the selected path fails, STP searches for and then establishes an alternate path to prevent or limit retransmission of data.

Enabling or Disabling STP

STP is disabled by default on the BigIron RX. Thus, new VLANs you configure on the BigIron RX have STP disabled by default. Table 10.1 lists the default STP states for the BigIron RX.

Table 10.1: Default STP States

Device Type	Default STP Type	Default STP State	Default STP State of New VLANs
BigIron RX	Foundry's multiple instances of spanning tree	Disabled	Disabled

By default, each VLAN on a BigIron RX runs a separate spanning tree instance. Each BigIron RX has one VLAN (VLAN 1) by default that contains all of its ports. However, if you configure additional port-based VLANs on a BigIron RX, 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 BigIron RX.

- 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 BigIron RX 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 BigIron RX, 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 BigIron RX 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 10.2 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 10.2: 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

Table 10.2: Default STP Bridge Parameters (Continued)

Parameter	Description	Default and Valid Values
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, Foundry 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 10.3 lists the default STP port parameters. The port parameters affect individual ports and are separately configurable on each port.

Table 10.3: 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 BigIron RX 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 10.2 on page 10-2.

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 10.3 on page 10-3. 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.

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 BigIron RX's Per VLAN Spanning Tree (PVST+) compatibility configuration. See "PVST/PVST+ Compatibility" on page 10-20.

The <num> parameter displays only the entries after the number you specify. For example, on a BigIron RX 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. See "Displaying Detailed STP Information for Each Interface" on page 10-8.

The **show spanning-tree** command shows the following information.

Table 10.4: 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.

Table 10.4: CLI Display of STP Information (Continued)

This Field...	Displays...
Bridge Parameters	
Bridge Identifier	The ID assigned by STP to this bridge for this spanning tree in hexadecimal. **Note**: If this address is the same as the Root ID, then this device or VLAN is the root bridge for its spanning tree.
Bridge MaxAge sec	The number of seconds this bridge waits for a hello message from the root bridge before deciding the root has become unavailable and performing a reconvergence.
Bridge Hello sec	The interval between each configuration BPDU sent by the bridge.
Bridge FwdDly sec	The number of seconds this bridge waits following a topology change and consequent reconvergence.
Hold Time sec	The minimum number of seconds that must elapse between transmissions of consecutive Configuration BPDUs on a port.
Last Topology Chang sec	The number of seconds since the last time a topology change occurred.
Topology Change cnt	The number of times the topology has changed since this device was reloaded.
Root Bridge Parameters	
Root Identifier	The ID assigned by STP to the root bridge for this spanning tree in hexadecimal.
Root Cost	The cumulative cost from this bridge to the root bridge. If this device is the root bridge, then the root cost is 0.
DesignatedBridge Identifier	The designated bridge to which the root port is connected. The designated bridge is the device that connects the network segment on the port to the root bridge.
Root Port	The port on this device that connects to the root bridge. If this device is the root bridge, then the value is “Root” instead of a port number.
Max Age sec	The number of seconds this root bridge waits for a hello message from the bridges before deciding a bridges has become unavailable and performing a reconvergence.
Hello sec	The interval between each configuration BPDU sent by the root bridge.
FwdDly sec	The number of seconds this root bridge waits following a topology change and consequent reconvergence.
Port STP Parameters	
Port Num	The port number.
Priority	The port’s STP priority. **Note**: If you configure this value, specify it in decimal format. See “Changing STP Port Parameters” on page 10-4.

Table 10.4: CLI Display of STP Information (Continued)

This Field...	Displays...
Path Cost	The port's STP path cost.
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 bridge(s) 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 BigIron RX 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 10.5: 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: • Foundry proprietary multiple Spanning Tree • IEEE 802.1Q Single Spanning Tree (SSTP) **Note**: If STP is disabled on a VLAN, the command displays the following message instead: "Spanning-tree of port-vlan <vlan-id> is disabled."
STP Bridge Parameters:	
Bridge identifier	The STP identity of this device.
Root	The ID assigned by STP to the root bridge for this spanning tree.
Control ports	The ports in the VLAN.
Active global timers	The global STP timers that are currently active, and their current values. The following timers can be listed: • Hello – The interval between Hello packets. This timer applies only to the root bridge. • Topology Change (TC) – The amount of time during which the topology change flag in Hello packets will be marked, indicating a topology change. This timer applies only to the root bridge. • Topology Change Notification (TCN) – The interval between Topology Change Notification packets sent by a non-root bridge toward the root bridge. This timer applies only to non-root bridges.
STP Port Parameters:	

Table 10.5: CLI Display of Detailed STP Information for Ports (Continued)

This Field...	Displays...
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 bridge(s) 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 BigIron RX runs a separate spanning tree, which you can enable or disable on an individual VLAN basis.

Alternatively, you can configure the BigIron RX to run a single spanning tree across all of its ports and VLANs. The SSTP feature is especially useful for connecting a BigIron RX 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 BigIron RX. See "Default STP Bridge and Port Parameters" on page 10-2.

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 BigIron RX places all the ports in a non-configurable VLAN, 4094, 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 BigIron RX 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 BigIron RX contains more than one port-based VLAN and you want all the ports to be in the same STP broadcast domain.

To configure the BigIron RX to run a single spanning tree, enter the following command at the global CONFIG level:

```
BigIron RX(config)# spanning-tree single
```

NOTE: If the BigIron RX 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, see “Default STP Port Parameters” on page 10-3.

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. See “Enabling or Disabling RSTP on a Single Spanning Tree” on page 11-28.

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, see “Displaying STP Information” on page 10-4.

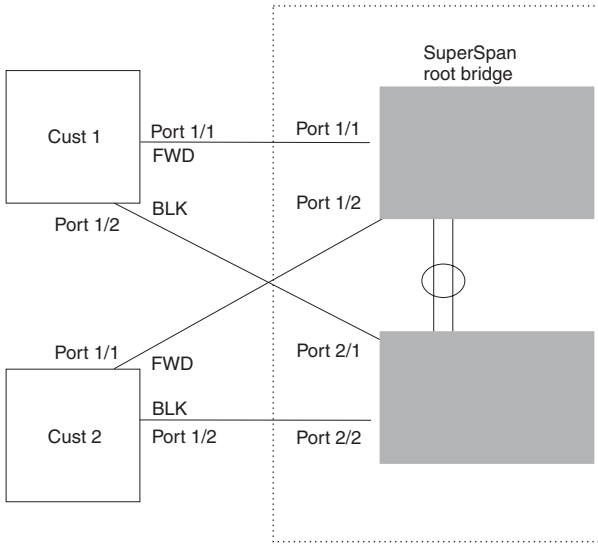
SuperSpan™

SuperSpan is a Foundry STP enhancement that allows Service Providers (SPs) to use STP in both SP networks and customer networks. The SP devices are BigIron RX 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 Foundry 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 10.1 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 BigIron RX devices in the SP are running SuperSpan.

Figure 10.1 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 BigIron RX devices in the SP. In Figure 10.1, the spanning trees of customer 1 and customer 2 do not interfere with one another because the SP network isolates each customer's spanning tree based on the SuperSpan customer IDs in the traffic.

BPDU Forwarding

When the BigIron RX receives a customer's BPDU on a boundary interface, the BigIron RX 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 BigIron RX that is configured for SuperSpan forwards the BPDU using the changed destination MAC address. At the other end of the tunnel, the BigIron RX 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

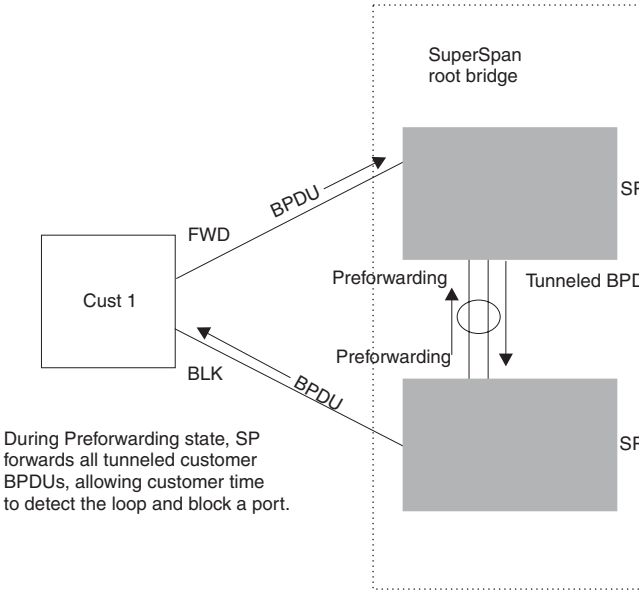
To ensure that the customer's network has time to converge at Layer 2 and prevent loops, the BigIron RX devices configured for SuperSpan use a special forwarding state, Preforwarding. The Preforwarding state occurs between the Learning and Forwarding states and by default lasts for five seconds. During the Preforwarding state, the

BigIron RX 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 Foundry 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 10.2 shows an example of how the Preforwarding state is used.

Figure 10.2 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 BigIron RX to a network that is already running SuperSpan, you must enable SuperSpan on the BigIron RX, at least on the VLANs that will be tunneling the customer traffic. Otherwise, the new BigIron RX does not use the Preforwarding state. This can cause the wrong ports to be blocked.

Combining Single STP and Multiple Spanning Trees

You can use SuperSpan in any of the following combinations:

- Customer and SP networks both use multiple spanning trees (a separate spanning tree in each VLAN).
- Customer uses multiple spanning trees but SP uses Single STP (all STP-enabled VLANs are in the same spanning tree).
- Customer uses Single STP but SP uses multiple spanning trees.
- Customer and SP networks both use Single STP.

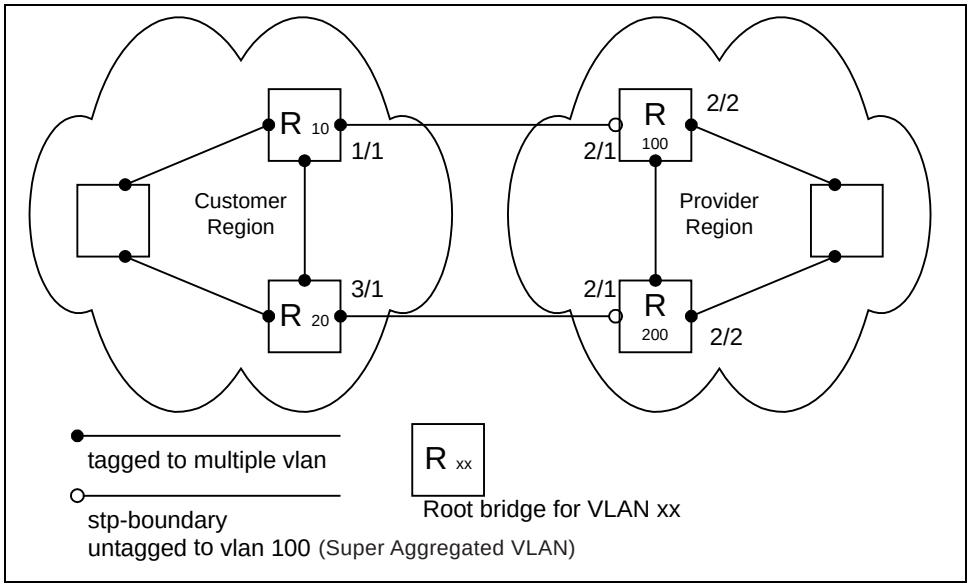
The following sections provide an example of each combination.

NOTE: All the combinations listed above are supported when the boundary ports joining the SP SuperSpan domain to the client spanning trees are untagged. For example, all these combinations are valid in super aggregated VLAN configurations. If the boundary ports are tagged, you cannot use Single STP in the client network in combination with multiple spanning trees in the SP SuperSpan domain.

Customer and SP Use Multiple Spanning Trees

Figure 10.3 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 10.3 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.

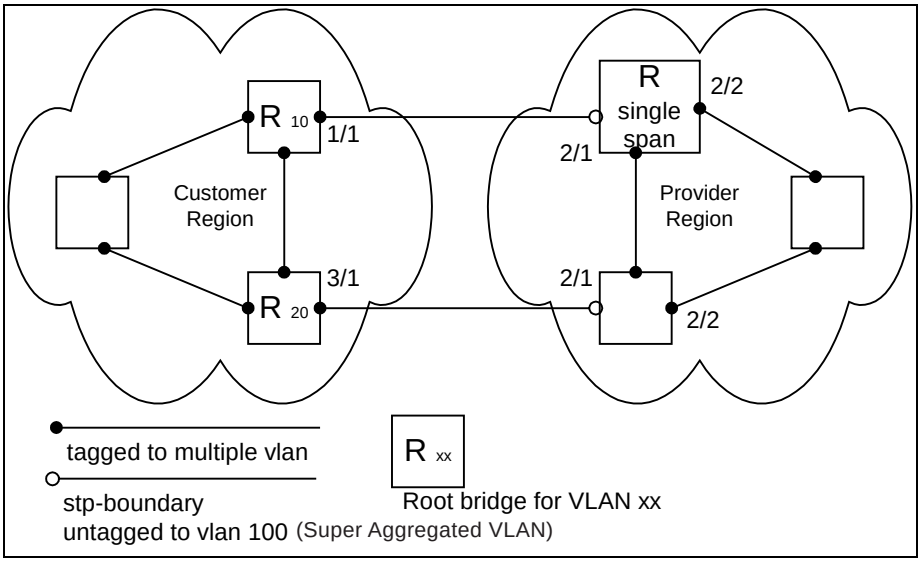
In the above example, STP in VLAN 10 will select R10 as the root bridge and make 1/1 on R10 forwarding while blocking port 3/1 on R20. The opposite occurs for STP in VLAN 20. As a result, both links connecting the customer and SP regions are fully utilized and serve as backup links at the same time, 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 10.4 shows an example of SuperSpan where the customer network uses multiple spanning trees while the SP network uses Single STP.

Figure 10.4 Customer using multiple spanning trees and SP using Single STP



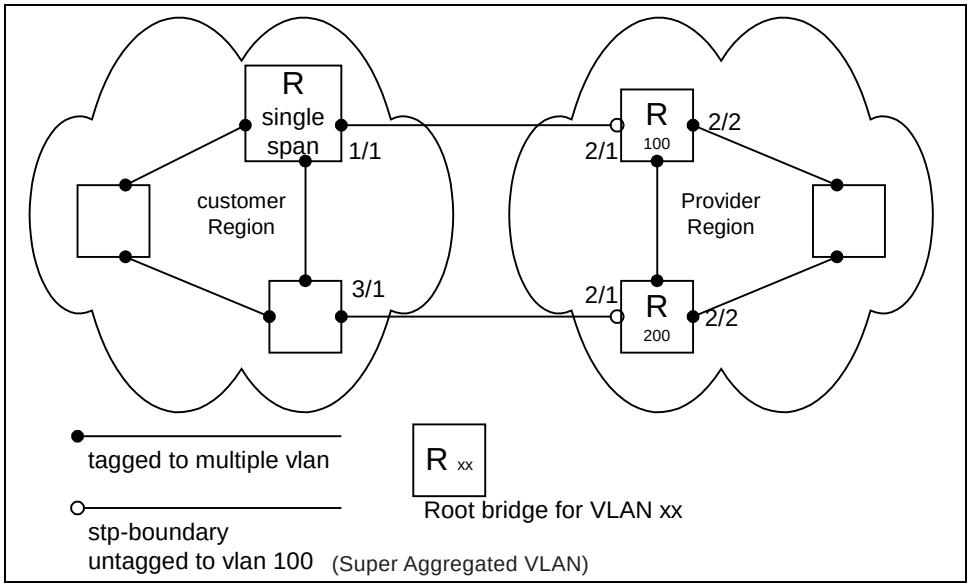
Customer traffic from different VLANs is maintained by different spanning trees, while the SP network is maintained by a single spanning tree. The SP can still use multiple VLANs at the core to separate traffic from different customers. However, all VLANs will have the same network topology because they 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 10-15, 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 10.5 shows an example of SuperSpan where the customer network uses Single STP while the SP uses multiple spanning trees.

Figure 10.5 Customer using Single STP and SP using multiple spanning trees

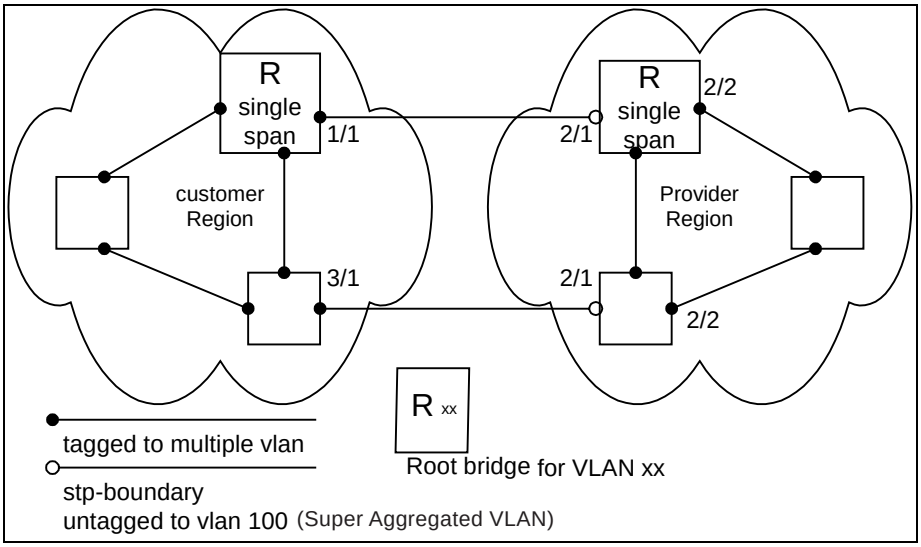


In this setup, the customer network is running a single spanning tree for VLANs 10 and 20. The traffic from VLAN 10 and 20 will be carried, or aggregated by VLAN 100 at the SP's network. The main difference between this scenario and the previous tow scenarios is that all traffic at the customer's network 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 10.6 shows an example of SuperSpan where the customer network and SP both use Single STP.

Figure 10.6 Customer and SP using Single STP



In this setup, both the customer and SP networks are running a single spanning tree at Layer 2. The traffic from VLAN 10 and 20 will be carried, or aggregated by VLAN 100 at the SP network as in the previous scenario. Loop-

free, non-blocking topology is still separately maintained by the customer's single spanning tree and the SP's single spanning tree.

Configuring SuperSpan

To configure the BigIron RX for SuperSpan:

- Configure each interface on the BigIron RX 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 Foundry network, the BigIron RX devices use the customer ID to ensure that BPDUs are forwarded only to the customer's devices, and not to other customers' devices.

- Globally enable SuperSpan. This step enables the Preforwarding state.

Configuring a Boundary Interface

To configure the boundary interfaces on SP 1 in Figure 10.1 on page 10-13, 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 BigIron RX as SuperSpan boundary interfaces. Interface 1/1 is a boundary interface with customer 1. Interface 1/2 is a boundary interface with customer 2. Each boundary interface is associated with a number, which is the SuperSpan ID. The SuperSpan ID identifies the instance of SuperSpan you are associating with the interface. Use the same SuperSpan ID for each boundary interface with the same customer. Use a different SuperSpan ID for each customer. For example, use SuperSpan ID 1 for all the boundary interfaces with customer 1 and use SuperSpan ID 2 for all boundary interfaces with customer 2.

Syntax: [no] stp-boundary <num>

The <num> parameter specifies the SuperSpan ID. Possible values: 1 – 65535.

To configure the boundary interfaces on SP 2 in Figure 10.1 on page 10-13, 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 globally enable SuperSpan, enter the following command:

```
BigIron RX(config)# super-span
```

Syntax: [no] super-span [preforward-delay <secs>]

The <secs> parameter specifies the length of the preforwarding state. You can specify from 3 – 15 seconds. The default is 5 seconds.

SuperSpan is enabled in all VLANs on the BigIron RX. 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  Customer    Tunnel
           BPDU Rx     BPDU Rx
     1/1   1            1
     1/2   0            0
     Total 1            1

CID 2 Boundary Ports:
     Port  Customer    Tunnel
           BPDU Rx     BPDU Rx
     2/1   0            3
     2/2   0            0
     Total 0            3
```

In this example, the BigIron RX 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 BigIron RX is shown.

This command shows the following information.

Table 10.6: CLI Display of SuperSpan Customer ID Information

This Field...	Displays...
CID	The SuperSpan customer ID number.
Port	The boundary port number.
Customer BPDU Rx	The number of BPDUs received from the client spanning tree.
Tunnel BPDU Rx	The number of BPDUs received from the SuperSpan tunnel.

To display general STP information, see “Displaying STP Information” on page 10-4.

PVST/PVST+ Compatibility

Foundry's support for Cisco's Per VLAN Spanning Tree plus (PVST+) allows the BigIron RX to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices[1]. Foundry ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected.

When it is configured for MSTP, the BigIron RX can interoperate with PVST.

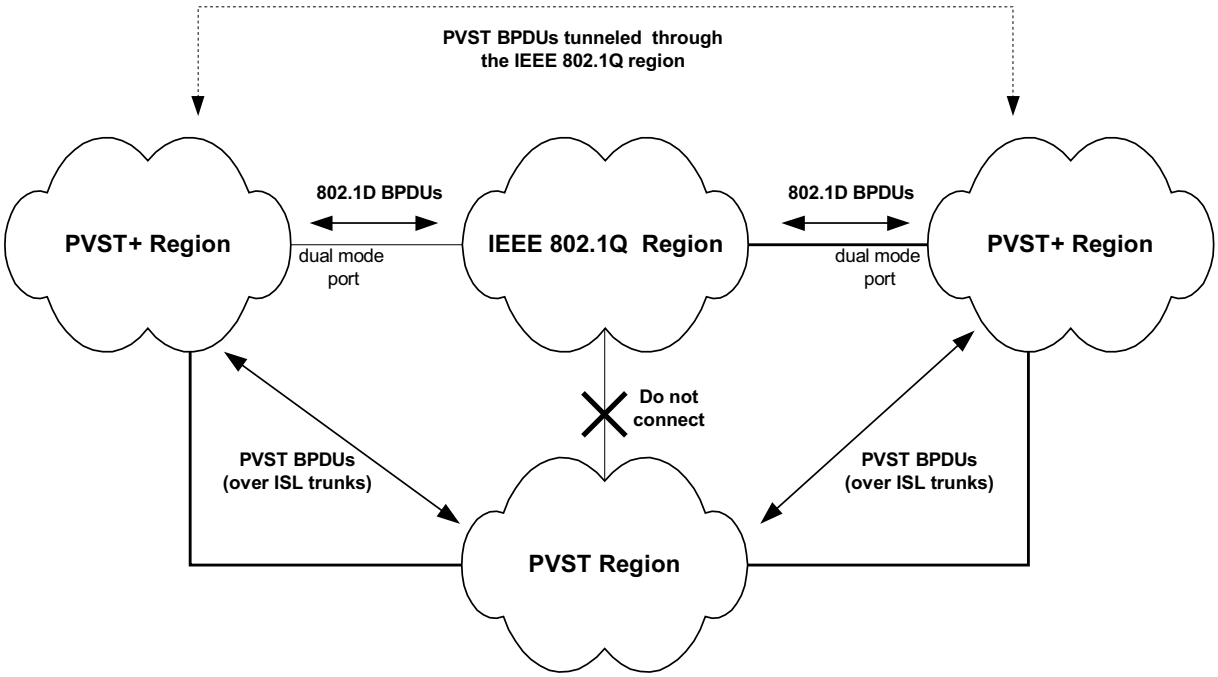
Overview of PVST and PVST+

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 BigIron RX 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 10.7 shows the interaction of IEEE 802.1Q, PVST, and PVST+ regions.

Figure 10.7 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+.

1.Cisco user documentation for PVST/PVST+ refers to the IEEE 802.1Q spanning tree as the ***Common Spanning Tree (CST)***.

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.

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 BigIron RX 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 BigIron RX.

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 BigIron RX, 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
```

1.Cisco PVST/PVST+ documentation refers to the Default VLAN as the ***Default Native VLAN***.

Syntax: show span pvst-mode

This command displays the following information.

Table 10.7: CLI Display of PVST+ Information

This Field...	Displays...
Port	The Foundry 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

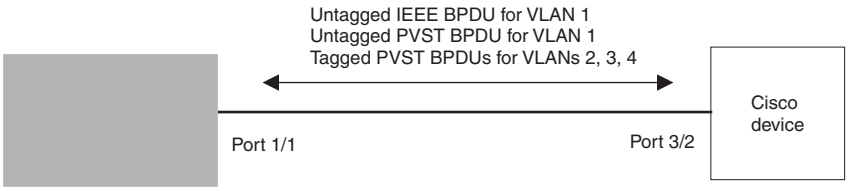
The examples use two common configurations:

- 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 10.8, a PVST+ configuration uses VLAN 1 as the untagged default VLAN and VLANs 2, 3, and 4 as tagged VLANs.

Figure 10.8 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.

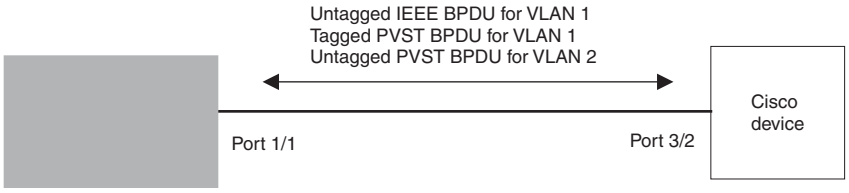
Port 1/1 will process BPDUs as follows:

- 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 10.9, 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 10.9 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.

Chapter 11
Configuring Rapid Spanning Tree Protocol

This chapter explains the IEEE 802.1W-2001Rapid Spanning Tree Protocols (RSTP) support on the BigIron RX.

IEEE 802.1W-2001 RSTP provides rapid traffic reconvergence for point-to-point links within a few milliseconds (< 500 milliseconds), following the failure of a bridge or bridge port.

This reconvergence occurs more rapidly than the reconvergence provided by the IEEE 802.1D Spanning Tree Protocol or by RSTP Draft 3 because:

- STP requires a newly selected Root port to go through listening and learning stages before traffic convergence can be achieved. The STP traffic convergence time is calculated using the following formula:

  2 x FORWARD_DELAY + BRIDGE_MAX_AGE.

- Convergence in RSTP bridges is not based on any timer values. Rather, it is based on the explicit handshakes between Designated ports and their connected Root ports to achieve convergence in less than 500 milliseconds.

NOTE: The rapid convergence will not occur on ports connected to shared media devices, such as hubs. To take advantage of the rapid convergence provided by RSTP, make sure to explicitly configure all point-to-point links in a topology.

Bridges and Bridge Port Roles

A bridge in an RSTP rapid spanning tree topology is assigned as the root bridge if it has the highest priority (lowest bridge identifier) in the topology. Other bridges are referred to as non-root bridges.

Unique roles are assigned to ports on the root and non-root bridges. Role assignments are based on the following information contained in the BPDU (RSTp packet):

- Root bridge ID
- Path cost value
- Transmitting bridge ID
- Designated port ID

RSTP algorithm uses this information to determine if the RST BPDU received by a port is superior to the RST BPDU that the port transmits. The two values are compared in the order as given above, starting with the Root bridge ID. The RST BPDU with a lower value is considered superior. The superiority and inferiority of the RST BPDU is used to assign a role to a port.

If the value of the received RST BPDU is the same as that of the transmitted RST BPDU, then the port ID in the RST BPDUs are compared. The RST BPDU with the lower port ID is superior. Port roles are then calculated appropriately.

The port's role is included in the BPDU that it transmits. The BPDU transmitted by an RSTP port is referred to as an RST BPDU, while it is operating in RSTP mode.

Ports can have one of the following roles:

- Root – Provides the lowest cost path to the root bridge from a specific bridge
- Designated – Provides the lowest cost path to the root bridge from a LAN to which it is connected
- Alternate – Provides an alternate path to the root bridge when the root port goes down
- Backup – Provides a backup to the LAN when the Designated port goes down
- Disabled – Has no role in the topology

Assignment of Port Roles

At system start-up, all RSTP-enabled bridge ports assume a Designated role. Once start-up is complete, RSTP algorithm calculates the superiority or inferiority of the RST BPDU that is received and transmitted on a port.

On a root bridge, each port is assigned a ***Designated port*** role, except for ports on the same bridge that are physically connected together. In these type of ports, the port that receives the superior RST BPDU becomes the ***Backup port***, while the other port becomes the ***Designated port***.

On non-root bridges, ports are assigned as follows:

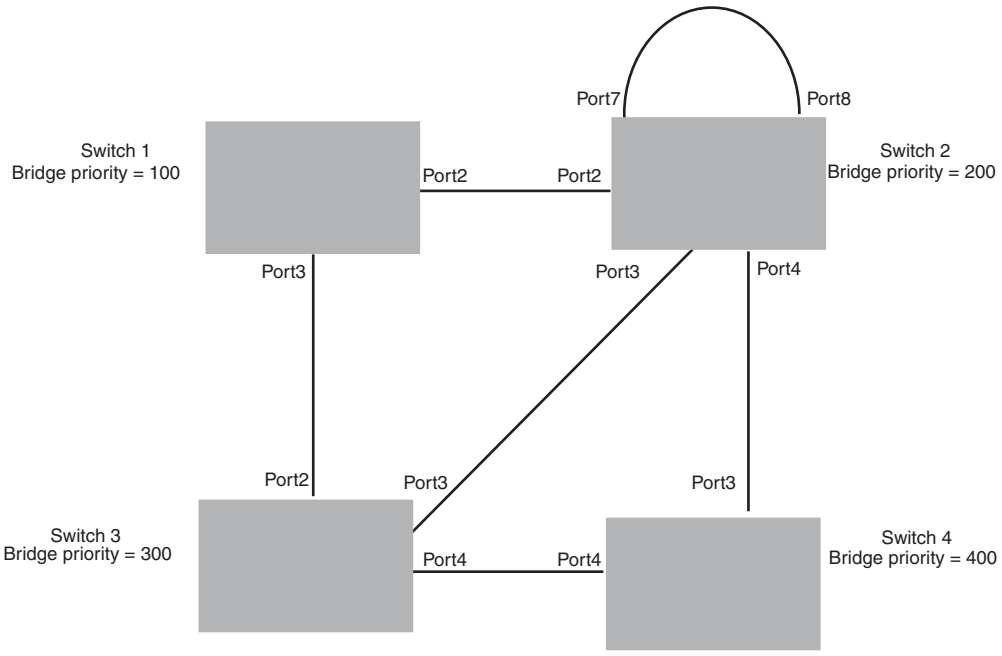
- The port that receives the RST BPDU with the lowest path cost from the root bridge becomes the ***Root port***.
- If two ports on the same bridge are physically connected, the port that receives the superior RST BPDU becomes the ***Backup port***, while the other port becomes the ***Designated port***.
- If a non-root bridge already has a Root port, then the port that receives an RST BPDU that is superior to those it can transmit becomes the ***Alternate port***.
- If the RST BPDU that a port receives is inferior to the RST BPDUs it transmits, then the port becomes a ***Designated port***.
- If the port is down or if RSTP is disabled on the port, that port is given the role of ***Disabled port***. Disabled ports have no role in the topology. However, if RSTP is enabled on a port with a link down and the link of that port comes up, then that port assumes one of the following port roles: Root, Designated, Alternate, or Backup.

The following example (Figure 11.1) 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 11.1 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 11.1 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

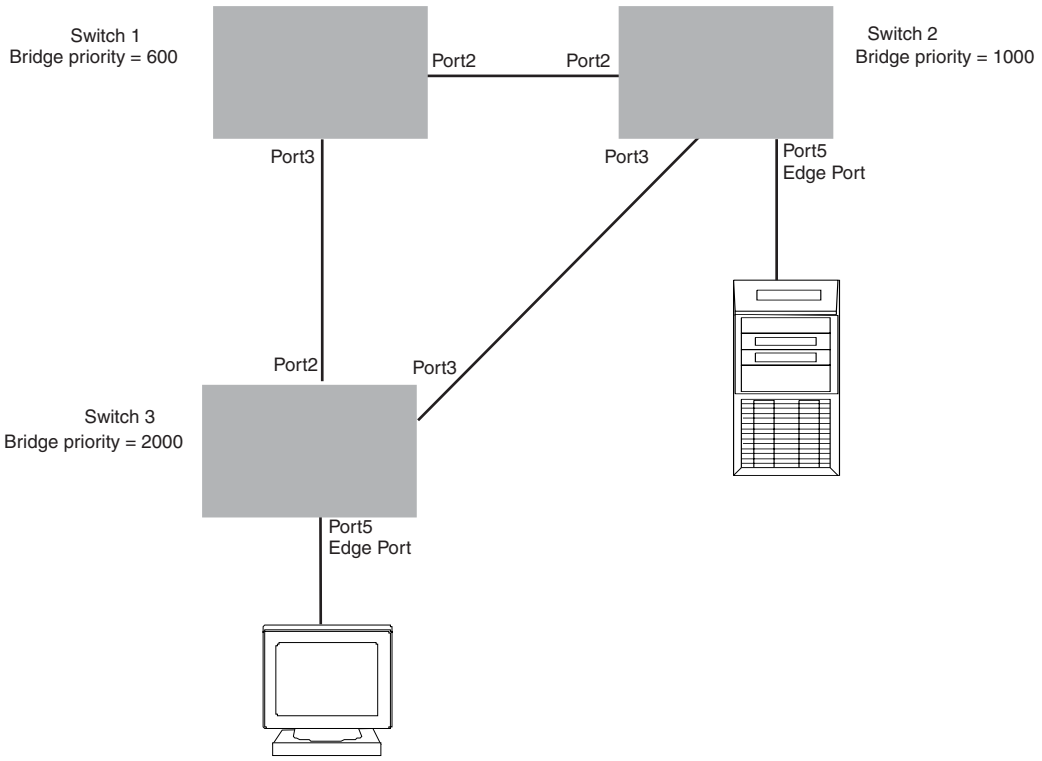
Switch 4 is not directly connected to the root bridge. It has two ports with superior incoming RST BPDUs from two separate LANs: Port3 and Port4. The RST BPDUs received on Port3 are superior to the RST BPDUs received on port 4; therefore, Port3 becomes the Root port and Port4 becomes the Alternate port.

Edge Ports and Edge Port Roles

Foundry's implementation of RSTP allows ports that are configured as Edge ports to be present in an RSTP topology. (Figure 11.2). 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 11.2 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.

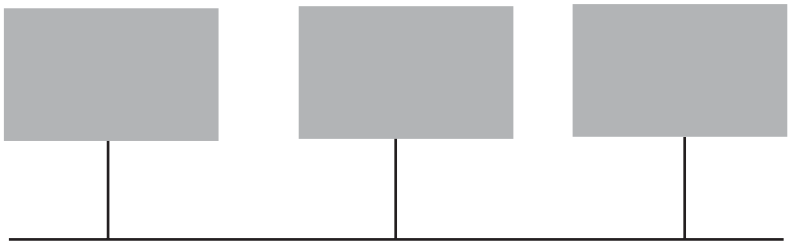
Point-to-Point Ports

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 11.3 is an example of shared media that should not be configured as point-to-point links. In Figure 11.3, a port on a bridge communicates or is connected to at least two ports.

Figure 11.3 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 11.4, 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

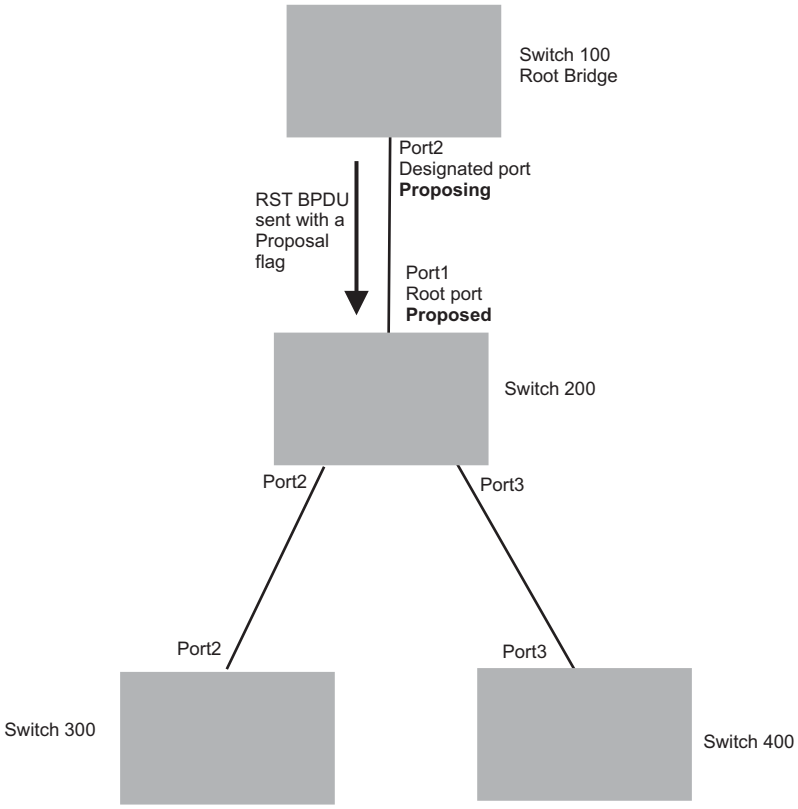
If a Root port has not been assigned on a bridge, RSTP uses the Proposing -> Proposed -> Sync -> Synced -> Agreed handshake:

- 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 11.4). The Designated port continues to send this flag in its RST BPDU until it is placed in a forwarding state (Figure 11.7) or is forced to operate in 802.1D mode. (See “Compatibility of RSTP with 802.1D” on page 27.)
- 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 11.4):
  - If the RST BPDU that the port receives is superior to what it can transmit, the port assumes the role of a Root port. (See the section on “Bridges and Bridge Port Roles” on page 11-1.)
  - 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.

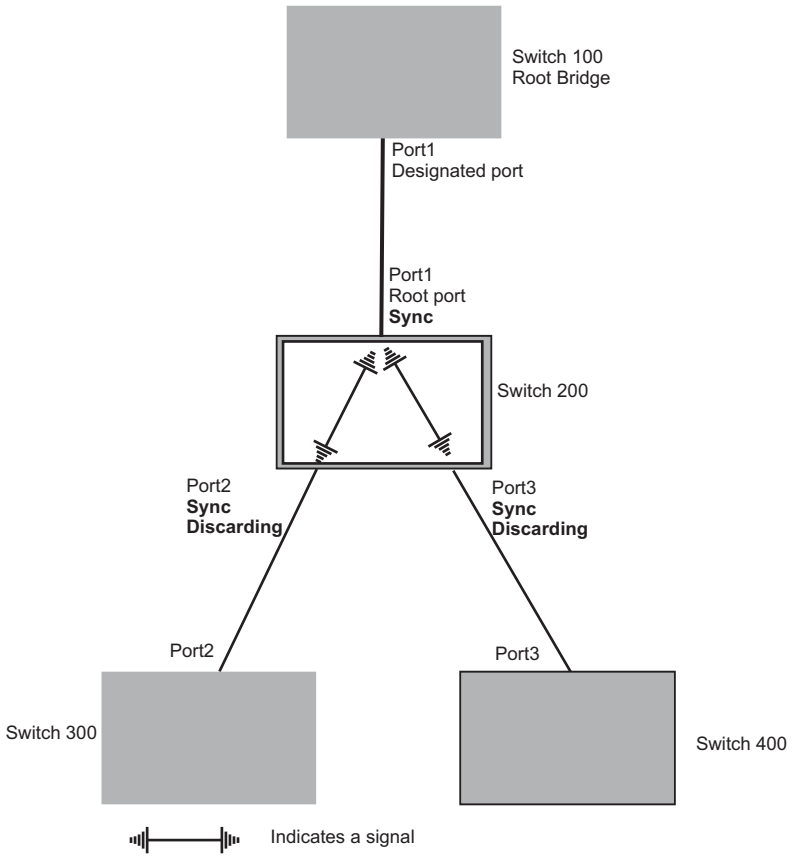
In Figure 11.4, Port3/Switch 200 is elected as the Root port

Figure 11.4 Proposing and Proposed Stage



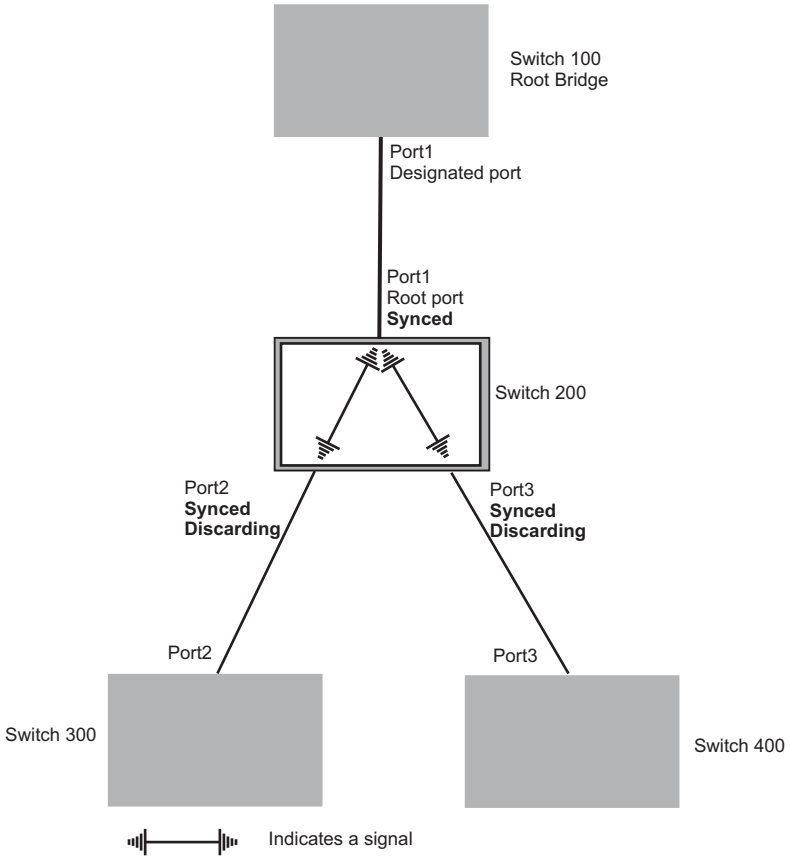
- Sync – Once the Root port is elected, it sets a sync signal on all the ports on the bridge. The signal tells the ports to synchronize their roles and states (Figure 11.5). Ports that are non-edge ports with a role of Designated port change into a discarding state. These ports have to negotiate with their peer ports to establish their new roles and states.

Figure 11.5 Sync Stage



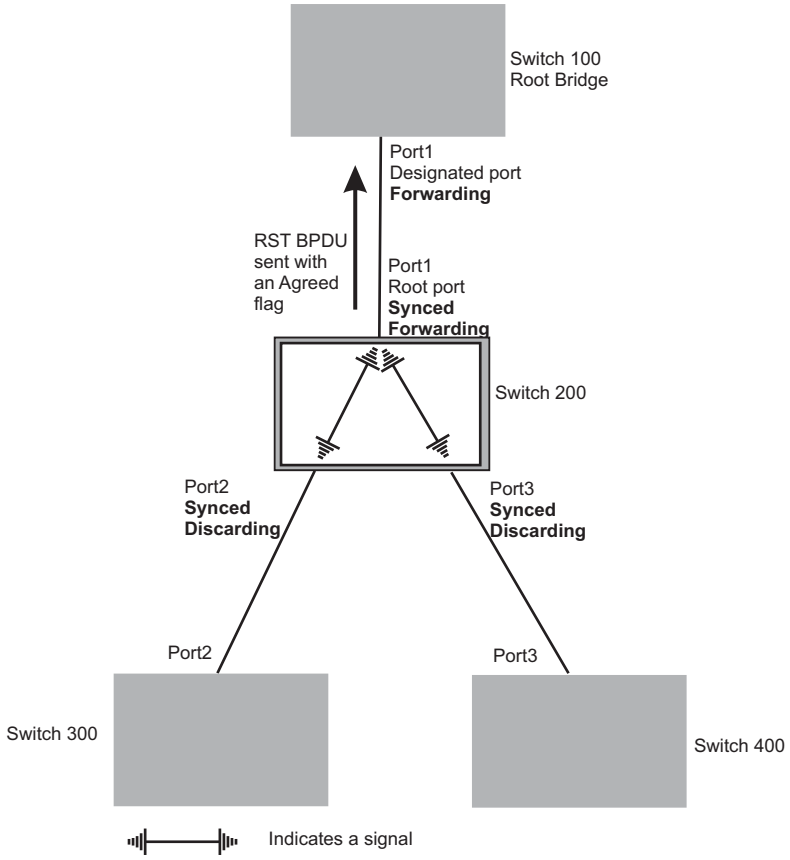
- Synced – Once the Designated port changes into a discarding state, it asserts a synced signal. Immediately, Alternate ports and Backup ports are synced. The Root port monitors the synced signals from all the bridge ports. Once all bridge ports asserts a synced signal, the Root port asserts its own synced signal (Figure 11.6).

Figure 11.6 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 11.7 Agree Stage



At this point, the handshake mechanism is complete between Switch 100, the root bridge, and Switch 200.

Switch 200 updates the information on the Switch 200's Designated ports (Port2 and Port3) and identifies the new root bridge. The Designated ports send RST BPDUs, containing proposal flags, to their downstream bridges, without waiting for the hello timers to expire on them. This process starts the handshake with the downstream bridges.

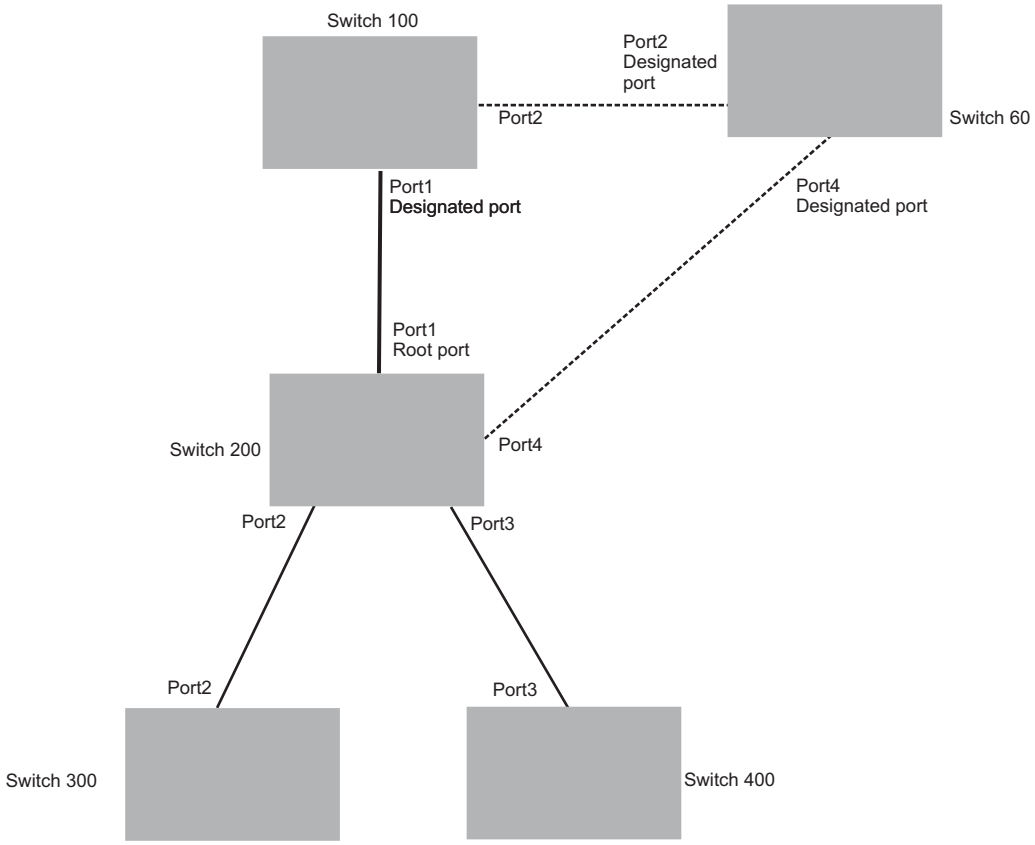
For example, Port2/Switch 200 sends an RST BPDU to Port2/Switch 300 that contains a proposal flag. Port2/Switch 300 asserts a proposed signal. Ports in Switch 300 then set sync signals on the ports to synchronize and negotiate their roles and states. Then the ports assert a synced signal and when the Root port in Switch 300 asserts it's synced signal, it sends an RST BPDU to Switch 200 with an agreed flag.

This handshake is repeated between Switch 200 and Switch 400 until all Designated and Root ports are in forwarding states.

Handshake When a Root Port Has Been Elected

If a non-root bridge already has a Root port, RSTP uses a different type of handshake. For example, in Figure 11.8, a new root bridge is added to the topology.

Figure 11.8 Addition of a New Root Bridge

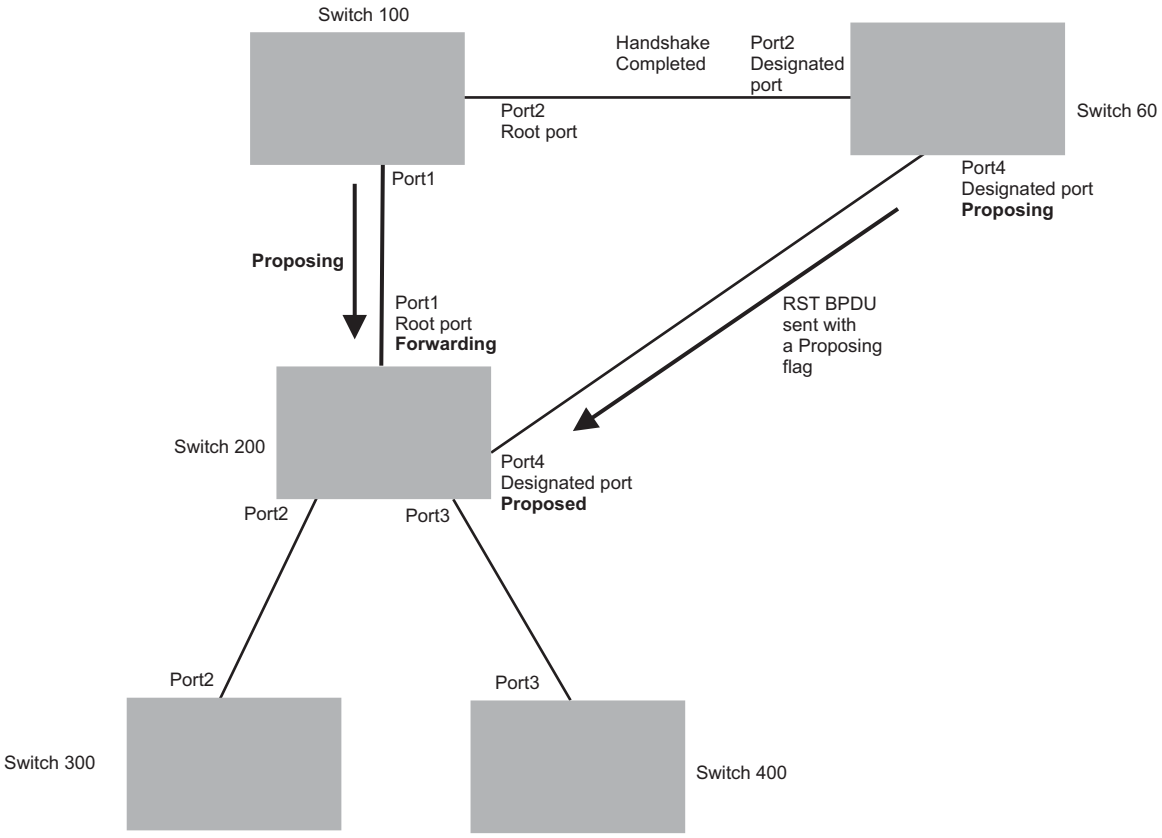


The handshake that occurs between Switch 60 and Switch 100 follows the one described in the previous section (“Handshake When No Root Port is Elected” on page 11-7). The former root bridge becomes a non-root bridge and establishes a Root port (Figure 11.9).

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:

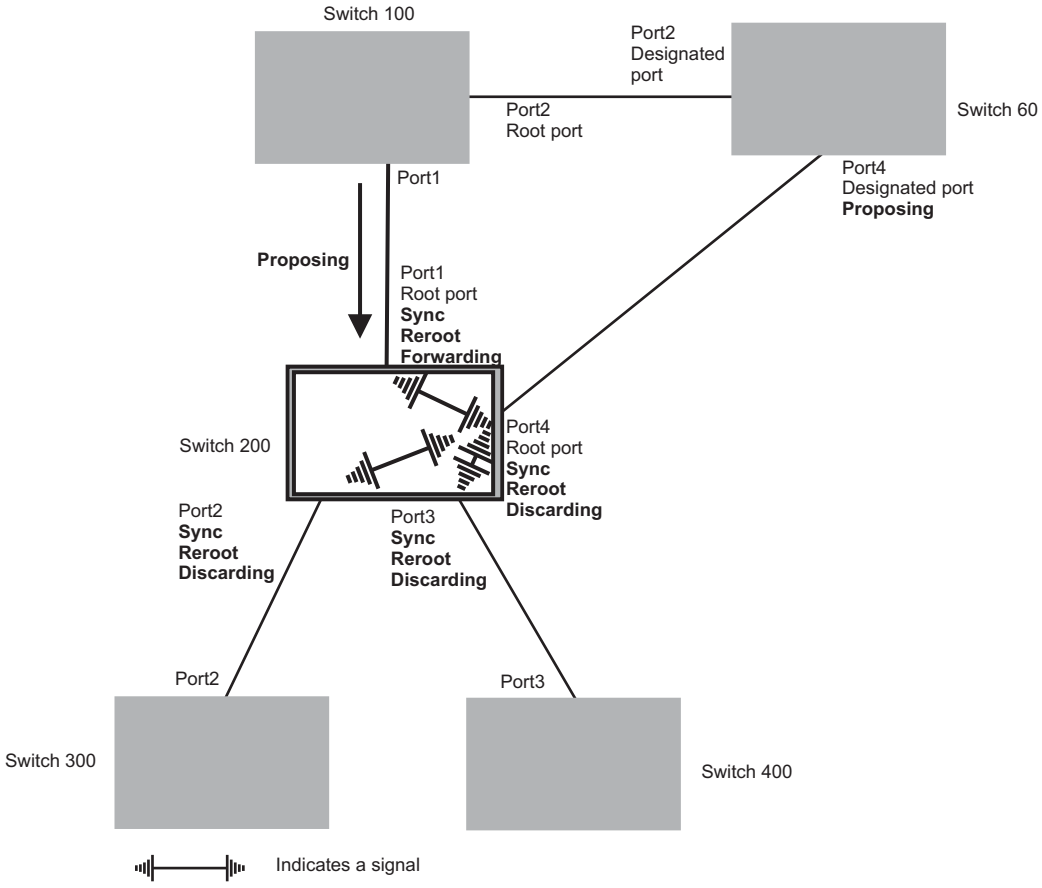
- Proposing and Proposed – The Designated port on the new root bridge (Port4/Switch 60) sends an RST BPDU that contains a proposing signal to Port4/Switch 200 to inform the port that it is ready to put itself in a forwarding state (Figure 11.9). RSTP algorithm determines that the RST BPDU that Port4/Switch 200 received is superior to what it can generate, so Port4/Switch 200 assumes a Root port role.

Figure 11.9 New Root Bridge Sending a Proposal Flag



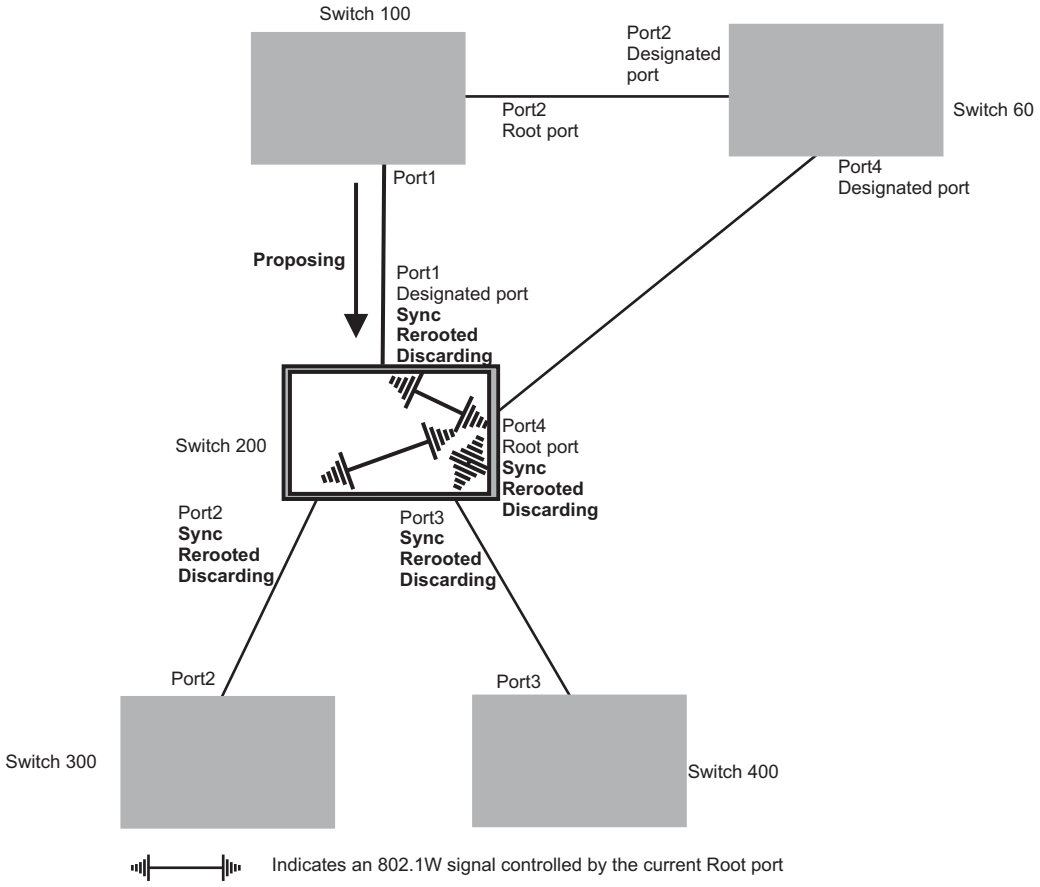
- Sync and Reroot – The Root port then asserts a sync and a reroot signal on all the ports on the bridge. The signal tells the ports that a new Root port has been assigned and they are to renegotiate their new roles and states. The other ports on the bridge assert their sync and reroot signals. Information about the old Root port is discarded from all ports. Designated ports change into discarding states (Figure 11.10).

Figure 11.10 Sync and Reroot



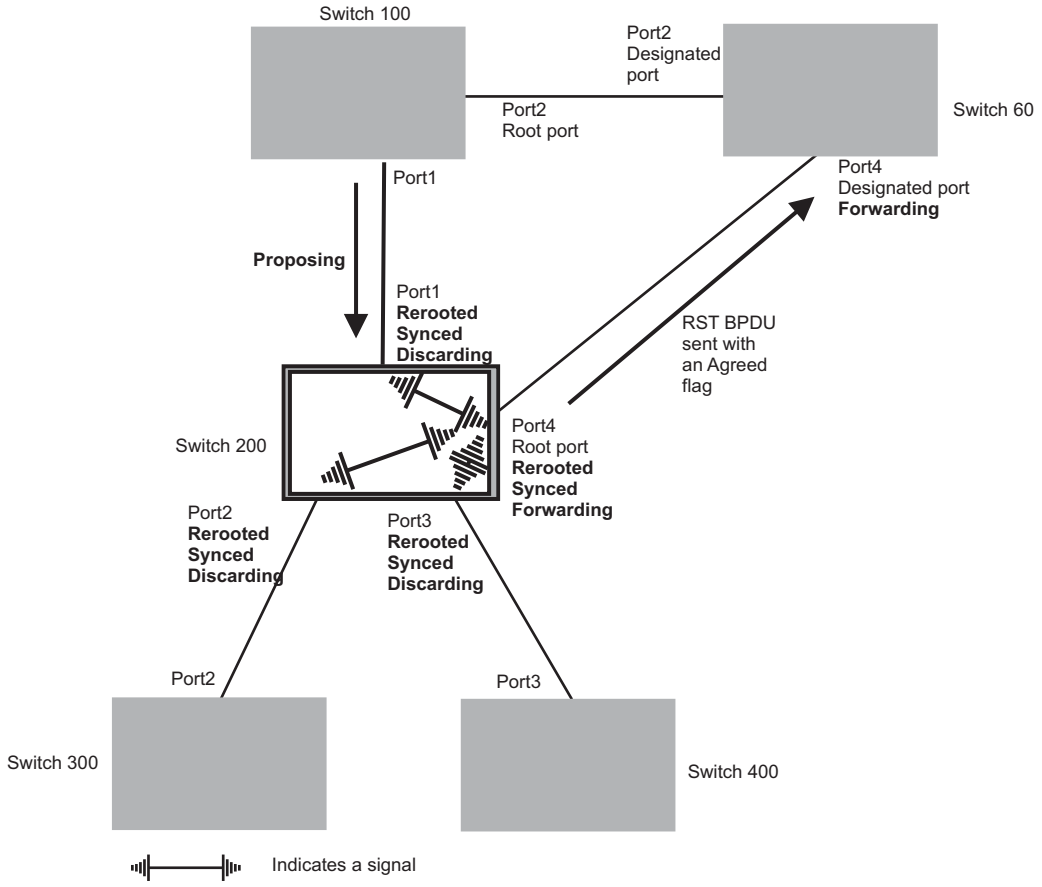
- Sync and Rerooted – When the ports on Switch 200 have completed the reroot phase, they assert their rerooted signals and continue to assert their sync signals as they continue in their discarding states. They also continue to negotiate their roles and states with their peer ports (Figure 11.11).

Figure 11.11 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 11.11). The Root port also moves into a forwarding state.

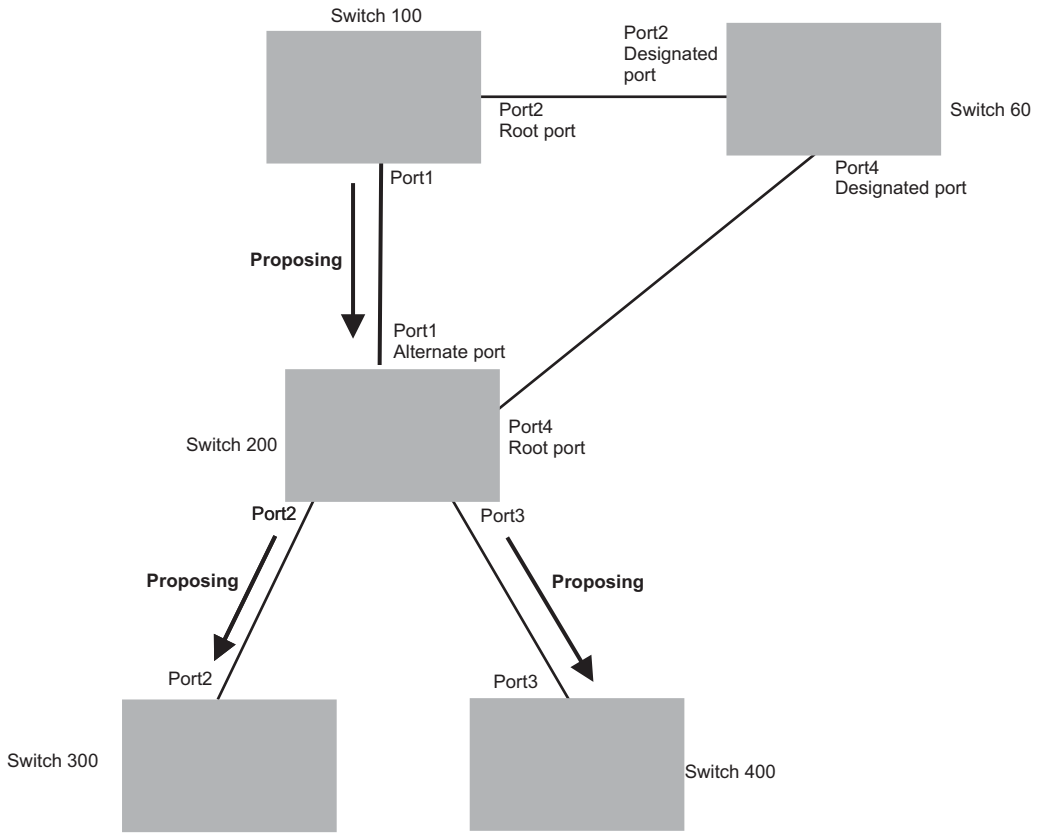
Figure 11.12 Rerooted, Synced, and Agreed



The old Root port on Switch 200 becomes an Alternate Port (Figure 11.13). 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 11.13 Handshake Completed After Election of New Root Port



Recall that Switch 200 sent the agreed flag to Port4/Switch 60 and not to Port1/Switch 100 (the port that connects Switch 100 to Switch 200). Therefore, Port1/Switch 100 does not go into forwarding state instantly. It waits until two instances of the forward delay timer expires on the port before it goes into forwarding state.

At this point the handshake between the Switch 60 and Switch 200 is complete.

The remaining bridges (Switch 300 and Switch 400) may have to go through the reroot handshake if a new Root port needs to be assigned.

Convergence in a Simple Topology

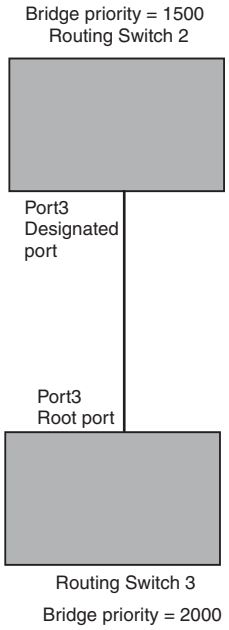
The examples in this section illustrate how RSTP convergence occurs in a simple Layer 2 topology at start-up.

NOTE: The remaining examples assume that the appropriate handshake mechanisms occur as port roles and states change.

Convergence at Start Up

In Figure 11.14, 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 11.14 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.

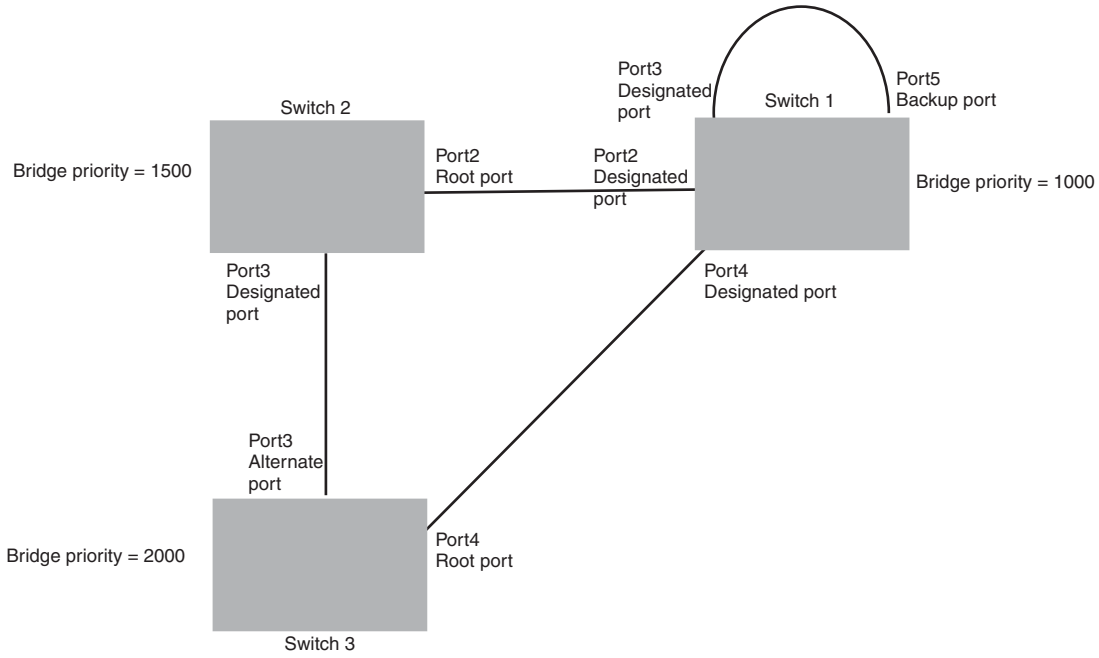
Port3/Switch 3, which starts with a role of Designated port, receives the RST BPDU and finds that it is superior to what it can transmit; therefore, Port3/Switch 3 assumes a new port role, that of a Root port. Port3/Switch 3 transmits an RST BPDU with an agreed flag back to Switch 2 and immediately goes 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 11.15).

Figure 11.15 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.

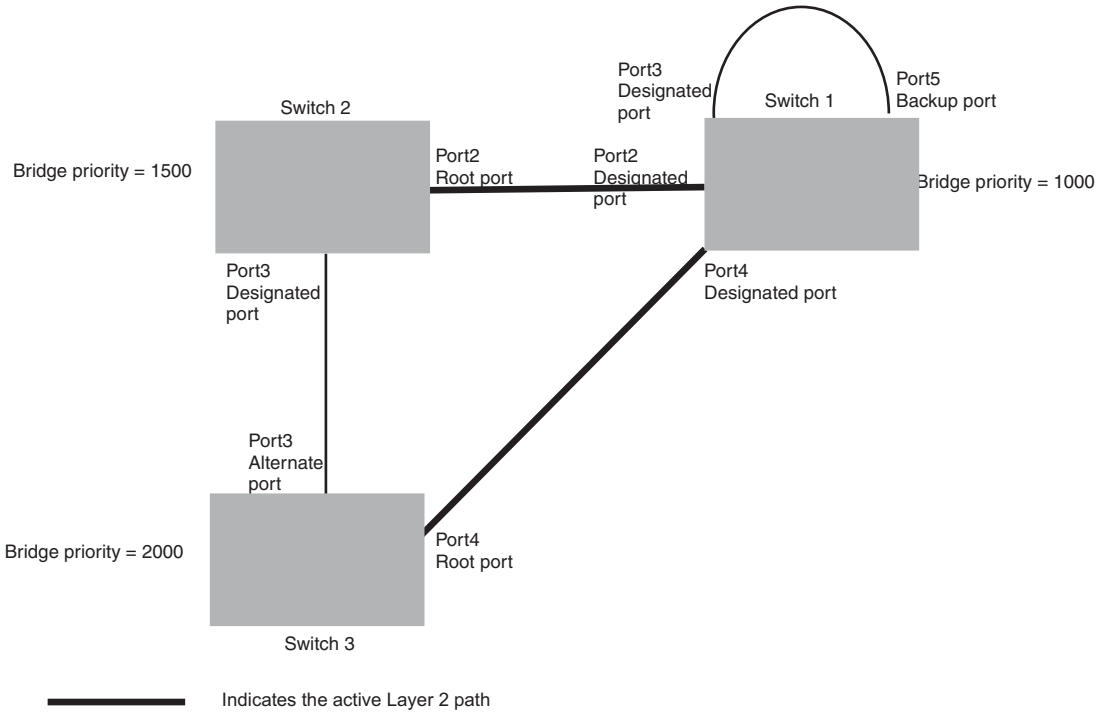
The Port2/Switch 2 bridge also sends an RST BPDU with an agreed flag Port2/Switch 1 that Port2 is the new Root port. Both ports go into forwarding states.

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 11.16.

Figure 11.16 Active Layer 2 Path

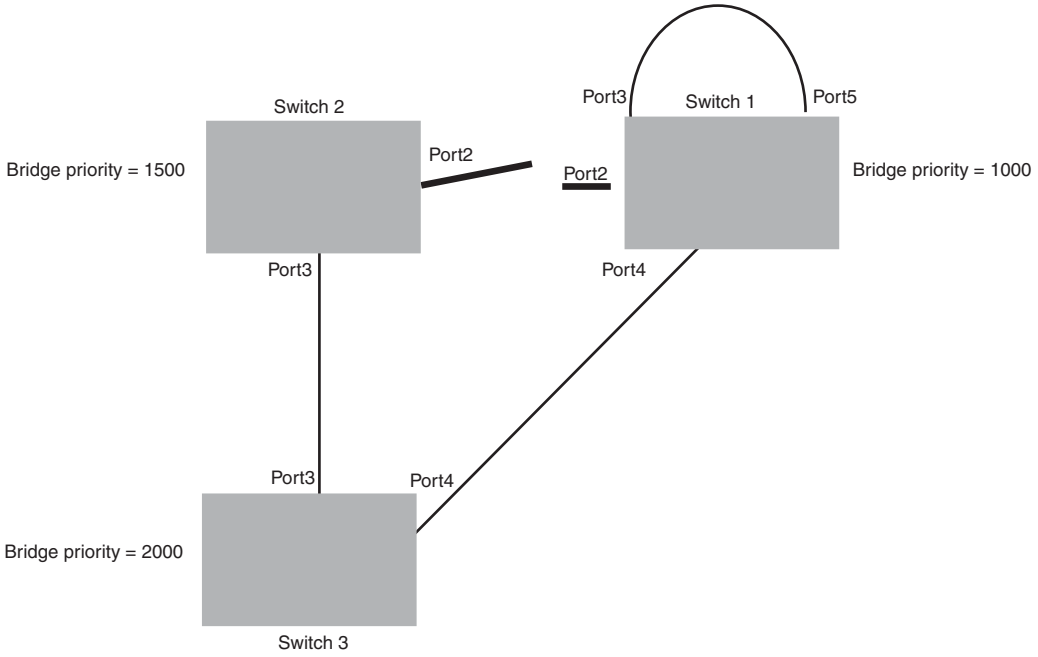


Convergence After a Link Failure

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 11.17).

Figure 11.17 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.

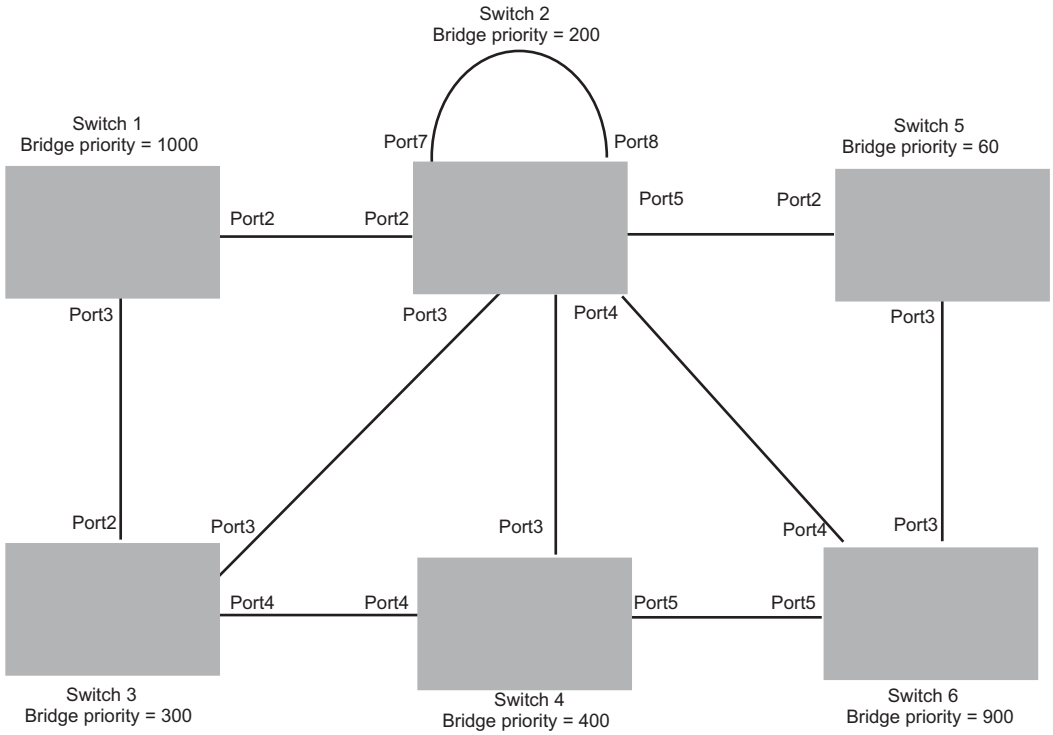
Now Port3/Switch 2 does not go into a forwarding state instantly like the Root port. It waits until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state. The wait, however, does not cause a denial of service, since the essential connectivity in the topology has already been established.

When fully restored, the topology is the same as that shown on Figure 11.15.

Convergence in a Complex RSTP Topology

The following is an example of a complex RSTP topology.

Figure 11.18 Complex RSTP Topology



In Figure 11.18, 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.

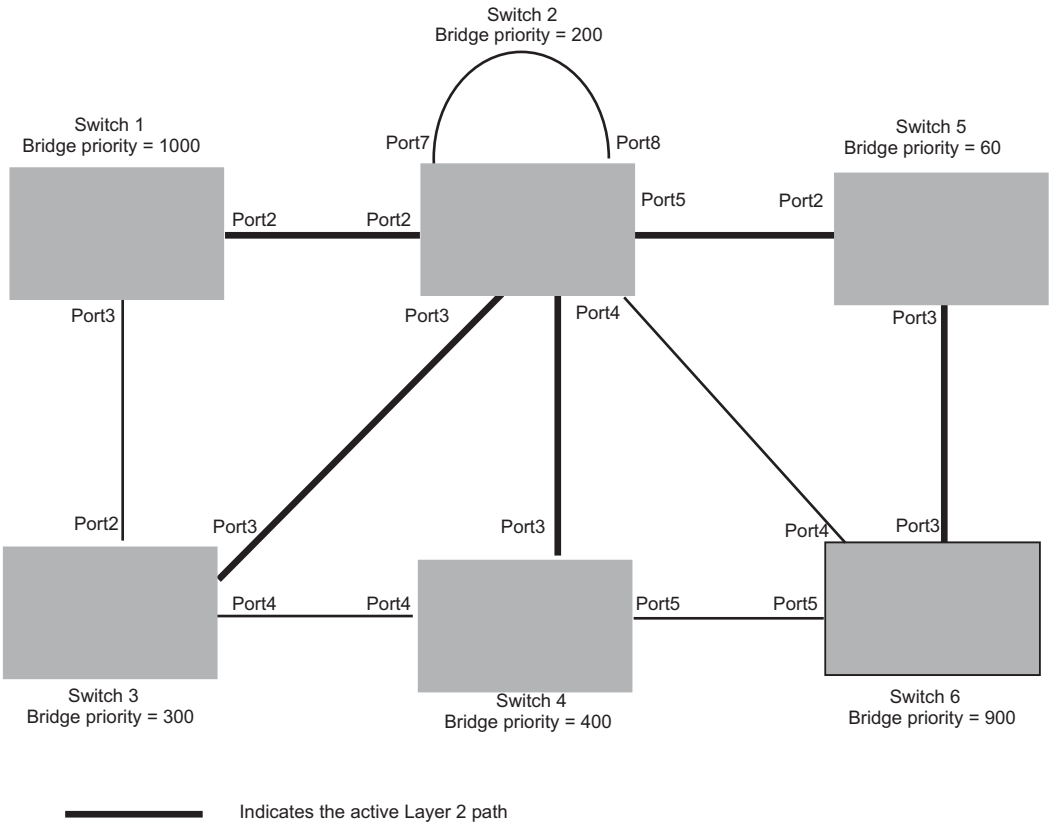
Port4/Switch 3 receives an RST BPDU that is inferior to what it can transmit; therefore, the port retains its Designated port role.

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 11.19 shows the active Layer 2 path of the topology in Figure 11.18.

Figure 11.19 Active Layer 2 Path in Complex Topology



Propagation of Topology Change

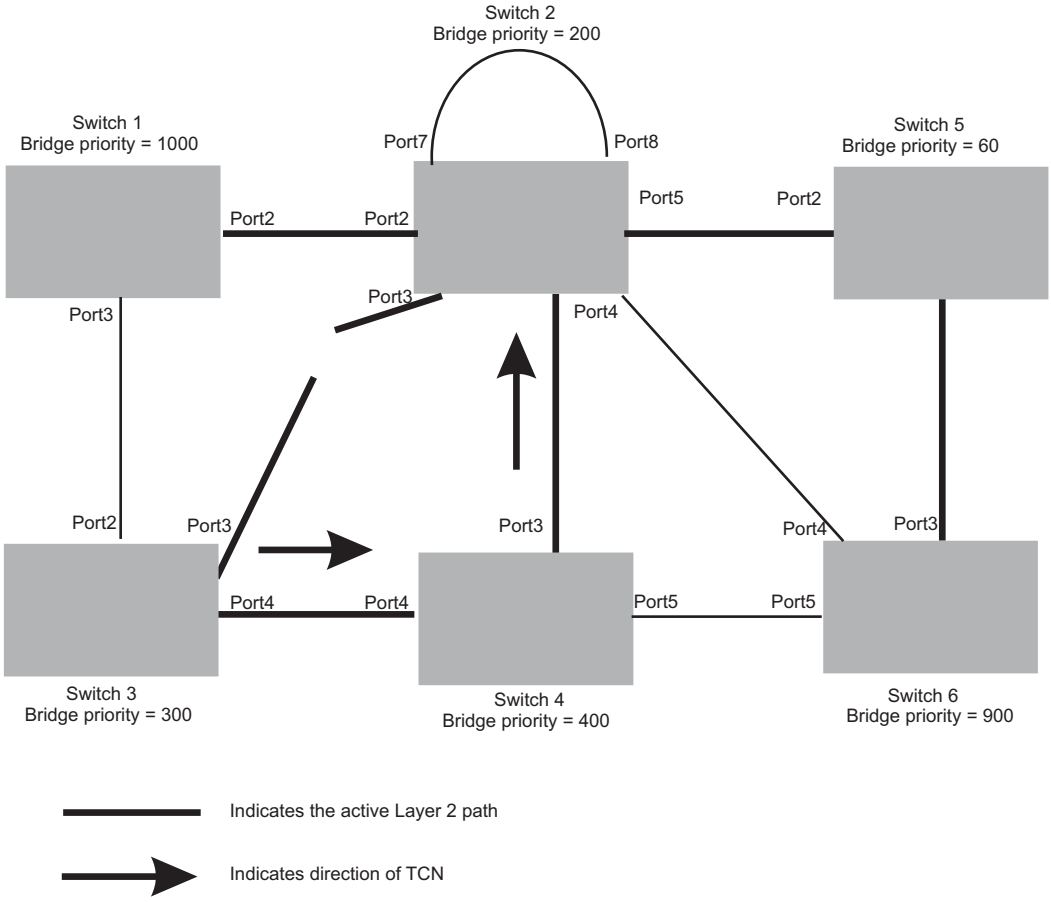
The Topology Change state machine generates and propagates the topology change notification messages on each port. When a Root port or a Designated port goes into a forwarding state, the Topology Change state machine on those ports send a topology change notice (TCN) to all the bridges in the topology 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 11.20, 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 11.20.)

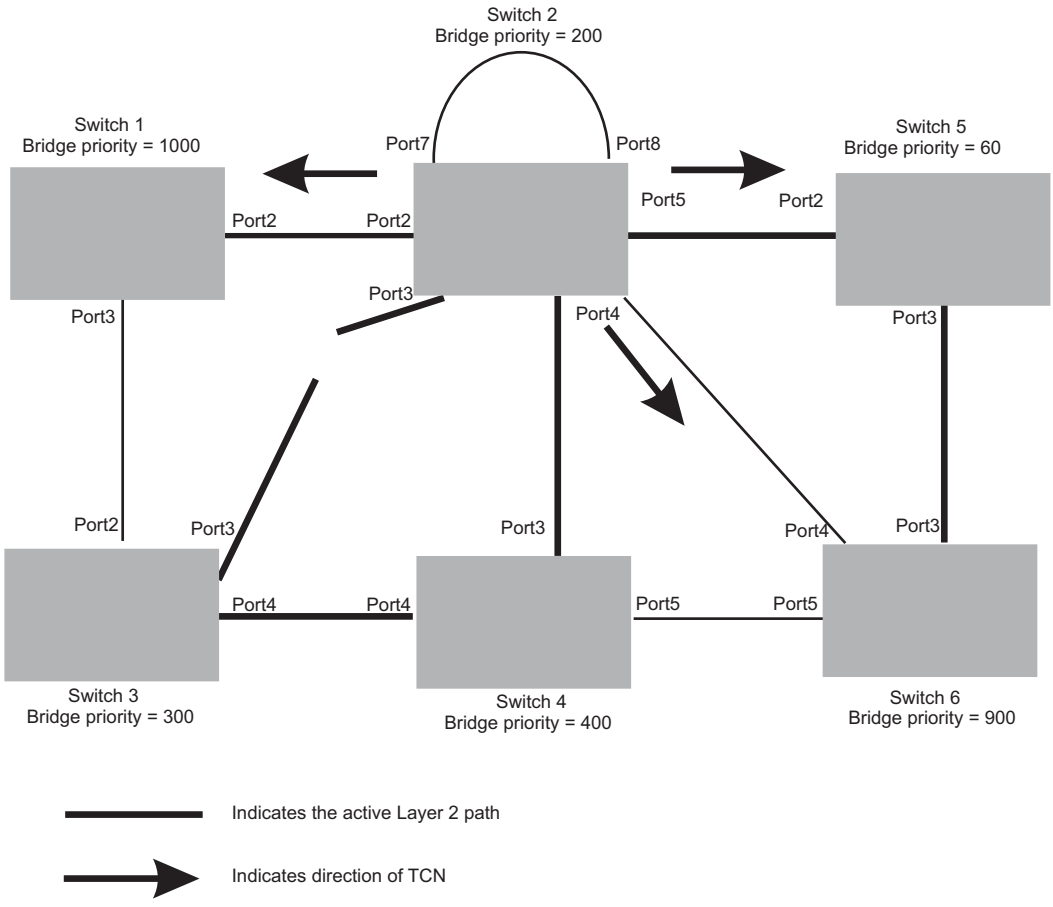
Figure 11.20 Beginning of Topology Change Notice



Switch 2 then starts the TCN timer on the Designated ports and sends RST BPDUs that contain the TCN as follows (Figure 11.21):

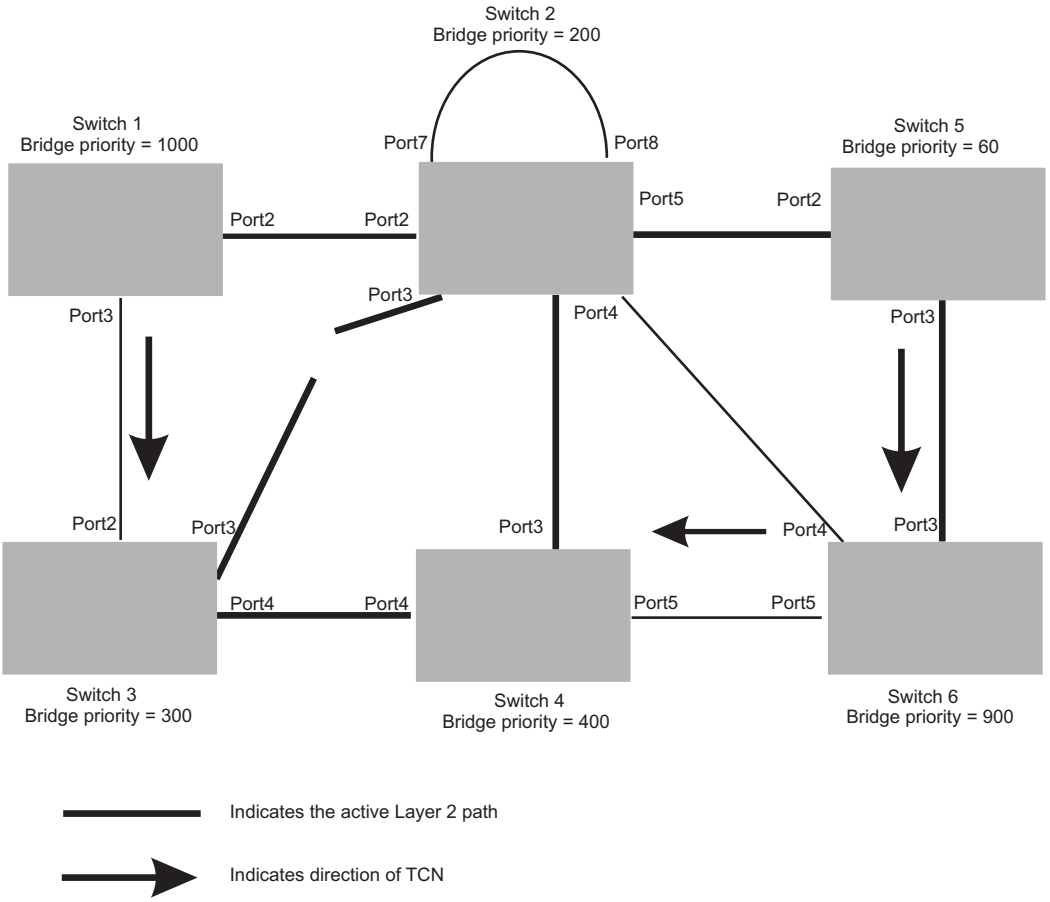
- Port5/Switch 2 sends the TCN to Port2/Switch 5
- Port4/Switch 2 sends the TCN to Port4/Switch 6
- Port2/Switch 2 sends the TCN to Port2/Switch 1

Figure 11.21 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 11.22).

Figure 11.22 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.**

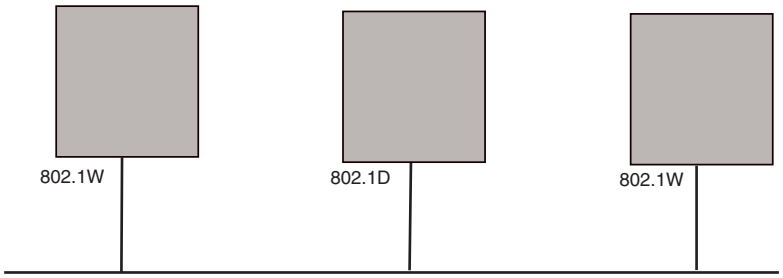
Compatibility with 802.1D means that an RSTP-enabled port can send BPDUs in the STP or 802.1D format when one of the following events occur:

- 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 11.23, 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 11.23 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 BigIron RX.

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 BigIron RX 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

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; however, set this value to at least 4 seconds to provide enough time for BPDUs to reach the root bridge before the timeout period expires on a non-root bridge port.

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 11.1 shows the recommended path cost values from the IEEE standards.

Table 11.1: Recommended Path Cost Values of RSTP

Link Speed	Recommended (Default) RSTP Path Cost Values	Recommended RSTP Path Cost Range
Less than 100 kilobits per second	200,000,000	20,000,000 – 200,000,000
1 Megabit per second	20,000,000	2,000,000 – 200,000,000
10 Megabits per second	2,000,000	200,000 – 200,000,000
100 Megabits per second	200,000	20,000 – 200,000,000
1 Gigabit per second	20,000	2,000 – 200,000,000
10 Gigabits per second	2,000	200 – 20,000
100 Gigabits per second	200	20 – 2,000
1 Terabits per second	20	2 – 200
10 Terabits per second	2	1 – 20

The **priority** <value> parameter specifies the preference that RSTP gives to this port relative to other ports for forwarding traffic out of the topology. You can specify a value from 0 – 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
```

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 11.2.

Table 11.2: 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.

Table 11.2: CLI Display of RSTP Summary (Continued)

This Field...	Displays...
Force-Version	The configured force version value. One of the following value is displayed: • 0 – The bridge has been forced to operate in an STP compatibility mode. • 2 – The bridge has been forced to operate in an RSTP mode. (This is the default.)
txHoldCnt	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Root Bridge Parameters:	
Root Bridge Identifier	ID of the Root bridge that is associated with this bridge
Root Path Cost	The cost to reach the root bridge from this bridge. If the bridge is the root bridge, then this parameter shows a value of zero.
Designated Bridge Identifier	The bridge from where the root information was received.It can be from the root bridge itself, but it could also be from another bridge.
Root Port	The port on which the root information was received. This is the port that is connected to the Designated Bridge.
Max Age	The ***max age*** is derived from the Root port. An RSTP-enabled bridge uses this value, along with the hello and message age parameters to compute the effective age of an RST BPDU. The ***message age*** parameter is generated by the Designated port and transmitted in the RST BPDU. RST BPDUs transmitted by a Designated port of the root bridge contains a message value of zero. ***Effective age*** is the amount of time the Root port, Alternate port, or Backup port retains the information it received from its peer Designated port. Effective age is reset every time a port receives an RST BPDU from its Designated port. If a Root port does not receive an RST BPDU from its peer Designated port for a duration more than the effective age, the Root port ages out the existing information and recomputes the topology. If the port is operating in 802.1D compatible mode, then max age functionality is the same as in 802.1D (STP).
Hello	The hello value derived from the Root port. It is the number of seconds between two Hello packets.

Table 11.2: CLI Display of RSTP Summary (Continued)

This Field...	Displays...
Fwd Dly	The number of seconds a non-edge Designated port waits until it can apply any of the following transitions, if the RST BPDU it receives does not have an agreed flag: • Discarding state to learning state • Learning state to forwarding state When a non-edge port receives the RST BPDU it goes into forwarding state within 4 seconds or after two hello timers expire on the port. Fwd Dly is also the number of seconds that a Root port waits for an RST BPDU with a proposal flag before it applies the state transitions listed above. If the port is operating in 802.1D compatible mode, then forward delay functionality is the same as in 802.1D (STP).
RSTP (IEEE 802.1W) Port Parameters	
Port Num	The port number shown in a slot#/port# format.
Pri	The configured priority of the port. The default is 128 or 0x80.
Port Path Cost	The configured path cost on a link connected to this port.
P2P Mac	Indicates if the point-to-point-mac parameter is configured to be a point-to-point link: • T – The link is configured as a point-to-point link. • F – The link is not configured as a point-to-point link. This is the default.
Edge port	Indicates if the port is configured as an operational Edge port: • T – The port is configured as an Edge port. • F – The port is not configured as an Edge port. This is the default.
Role	The current role of the port: • Root • Designated • Alternate • Backup • Disabled Refer to “Bridges and Bridge Port Roles” on page 11-1 for definitions of the roles.

Table 11.2: CLI Display of RSTP Summary (Continued)

This Field...	Displays...
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 11-5 and "Edge Port and Non-Edge Port States" on page 11-5.
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.

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

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 11-1 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 11-5 and “Edge Port and Non-Edge Port States” on page 11-5.

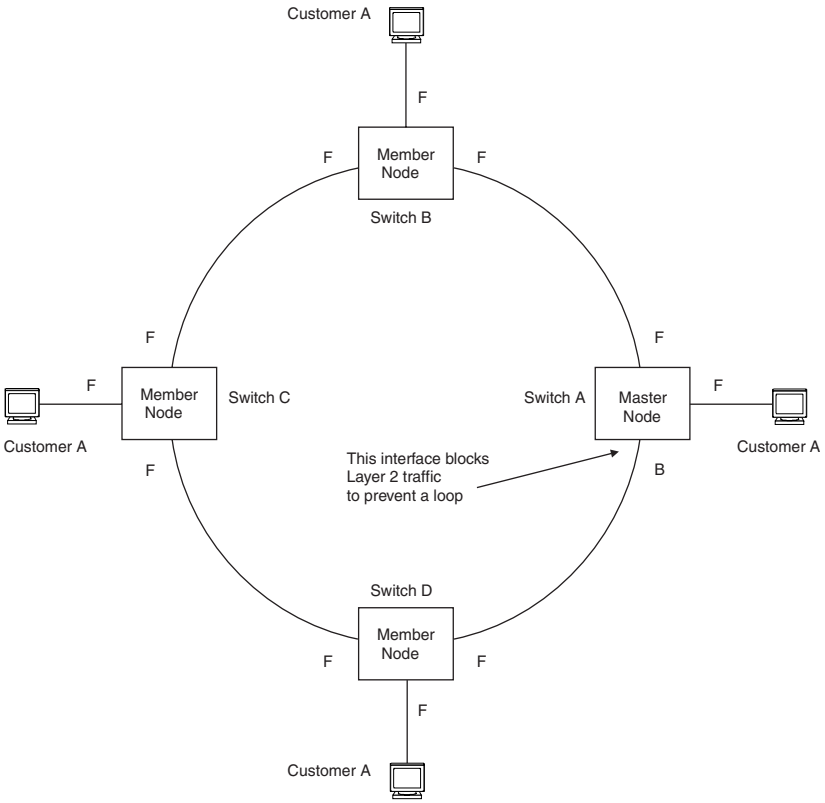
Chapter 12
Metro Ring Protocol (MRP)

MRP is a Foundry 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 12.1 shows an MRP metro ring.

Figure 12.1 Metro ring – normal state



The ring in this example consists of four MRP nodes (Foundry 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 loops.

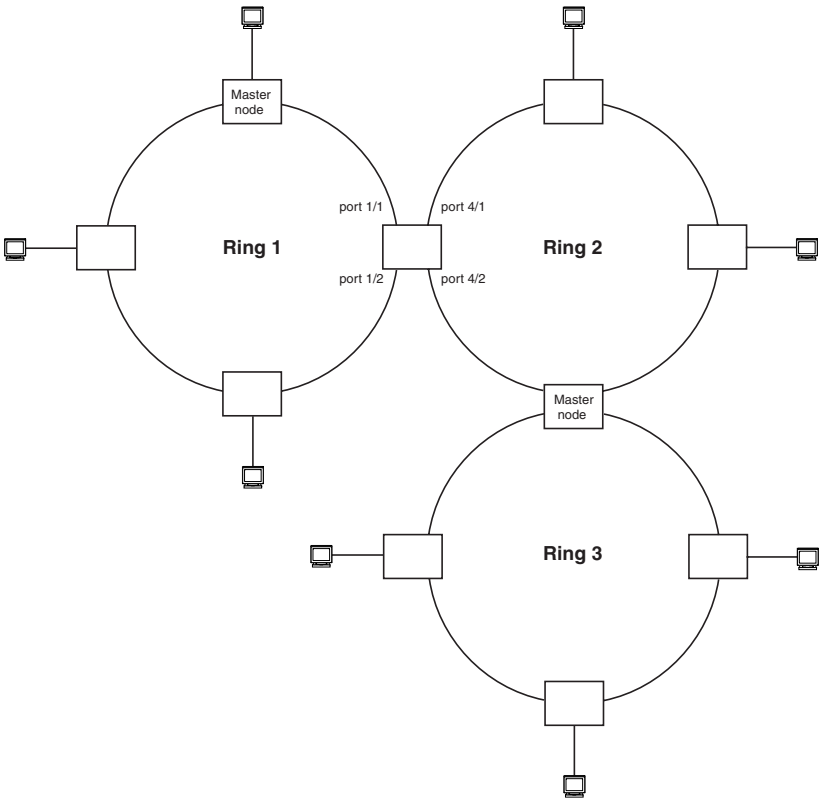
NOTE: When you configure MRP, Foundry recommends that you disable one of the ring interfaces before beginning the ring configuration. Disabling an interface prevents a Layer 2 loop from occurring while you are configuring MRP on the ring nodes. Once MRP is configured and enabled on all the nodes, you can re-enable the interface.

MRP Rings Without Shared Interfaces (MRP Phase 1)

MRP Phase 1 allows you to configure multiple MRP rings, as shown in Figure 12.2, 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. (See Figure 12.2.) Each ring is an independent ring and RHP packets are processed within each ring.

Figure 12.2 Metro ring – multiple rings

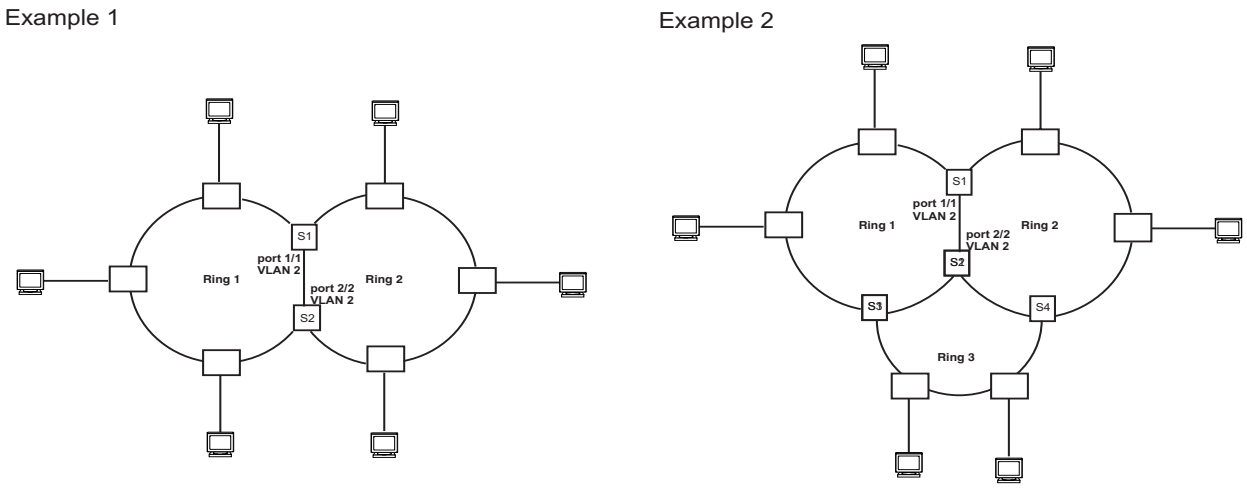


In this example, two nodes are each configured with two MRP rings. Any node in a ring can be the master for its ring. A node also can be the master for more than one ring.

MRP Rings with Shared Interfaces (MRP Phase 2)

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 12.3 shows multiple MRP rings that share the same interface.

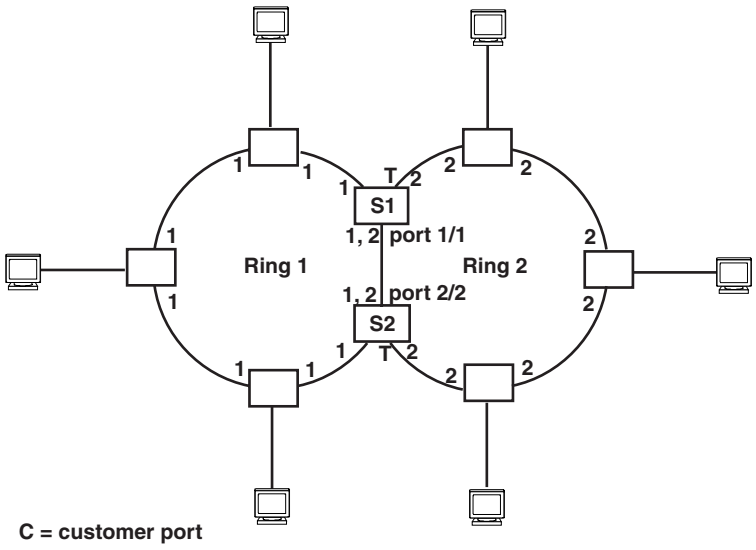
Figure 12.3 Examples of multiple rings sharing the same interface



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.

Figure 12.4 Interface IDs and Types on Rings with Shared Interfaces



For example, in Figure 12.4, 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 12.4, 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 12.4, 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.

Selection of Master Node on Shared Interfaces

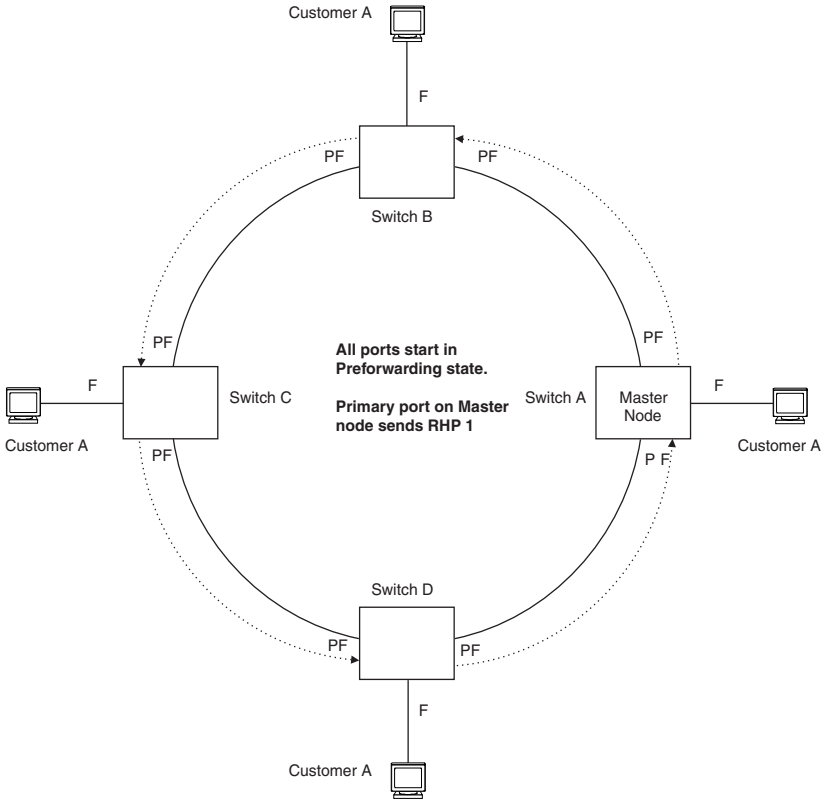
Allowing MRP rings to share interfaces limits the nodes that can be designated as the master node. Any node on an MRP ring that does not have a shared interface can be designated as the ring's master node. However, if all nodes on the ring have shared interfaces, nodes that do not have tunnel ports can 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 12.4, 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.

Ring Initialization

The ring shown in Figure 12.1 shows the port states in a fully initialized ring without any broken links. Figure 12.5 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 12.5 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.

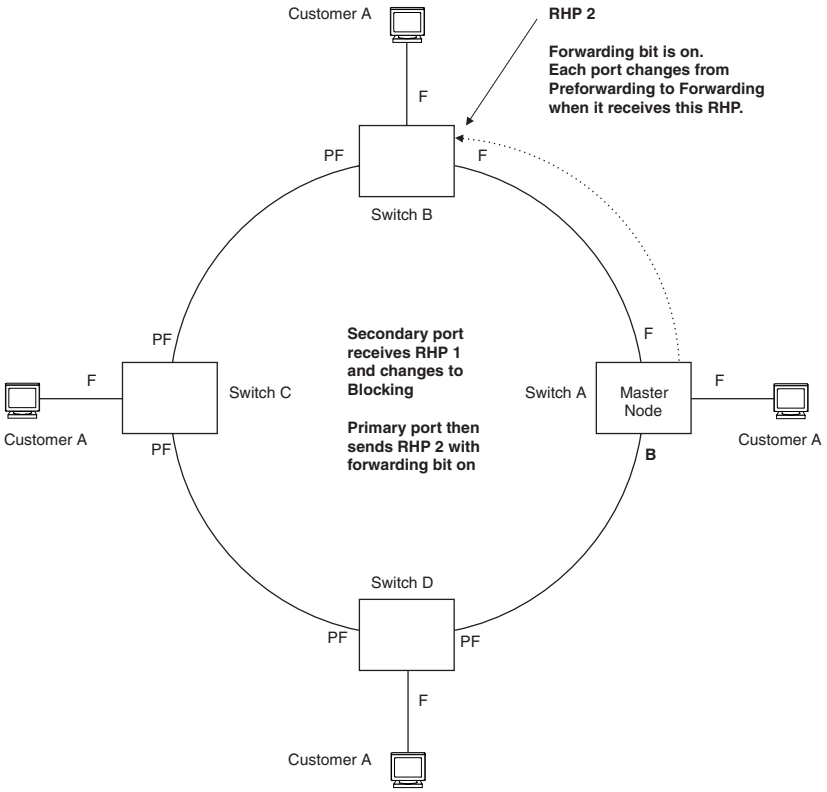
When MRP is enabled, all ports begin in the Preforwarding state. The primary interface on the Master node, although it is in the Preforwarding state like the other ports, immediately sends an RHP onto the ring. The secondary port on the Master node listens for the RHP.

- 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 12.6 shows an example.

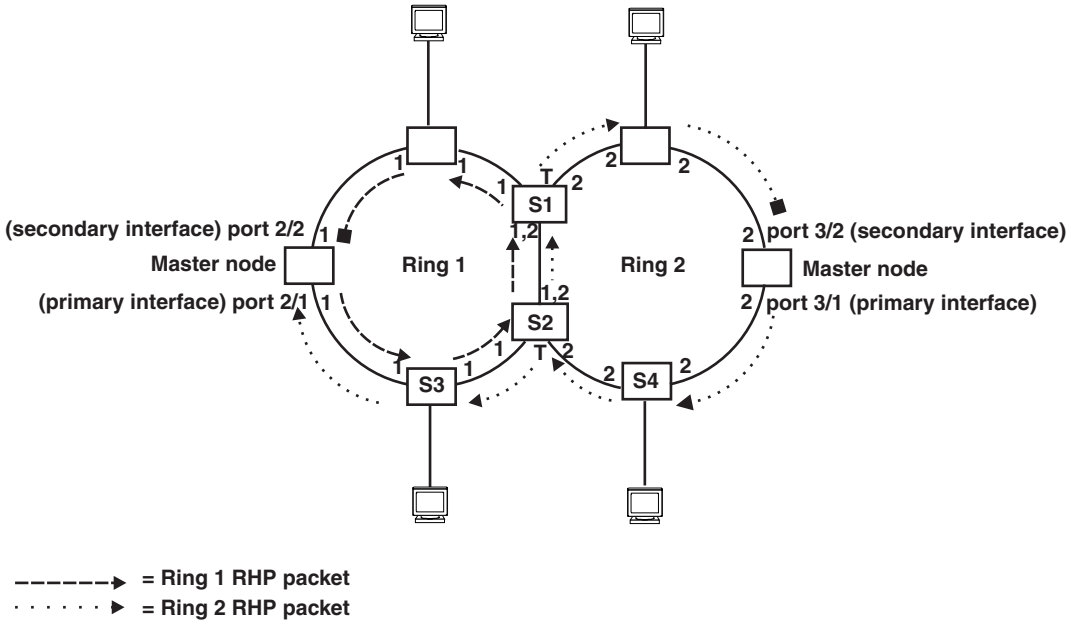
Figure 12.6 Metro ring – from Preforwarding to Forwarding



Each RHP also has a sequence number. MRP can use the sequence number to determine the round-trip time for RHPs in the ring. See “Using MRP Diagnostics” on page 12-12.

Figure 12.7 shows an example of how RHP packets are processed normally in MRP rings with shared interfaces.

Figure 12.7 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.

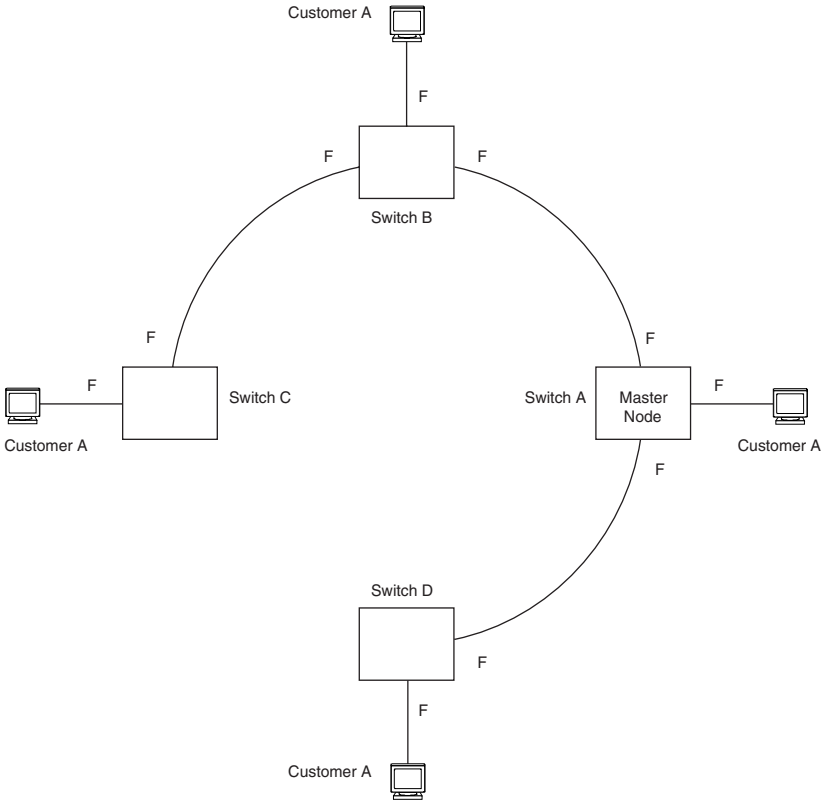
When the RHP packet reaches the interface on node S2 shared by Rings 1 and 2, the packet is forwarded since its priority is less than the interface's priority. The packet continues to be forwarded to node S1 until it reaches the tunnel port on S1. That tunnel port determines that the RHP packet's priority 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.

How Ring Breaks Are Detected and Healed

Figure 12.8 shows ring interface states following a link break. MRP quickly heals the ring and preserves connectivity among the customer networks.

Figure 12.8 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 12.8.
- Forwarding interfaces – Each member interface remains in the Forwarding state.

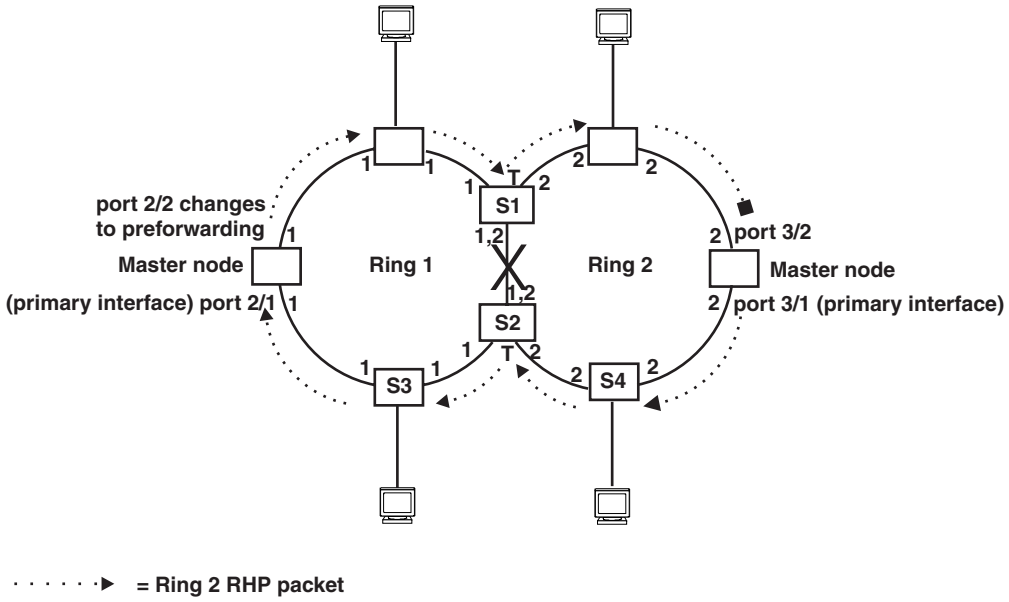
When the broken link is repaired, the link's interfaces come up in the Preforwarding state, which allows RHPs to travel through the restored interfaces and reach the secondary interface on the Master node.

- 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.

If the link between ***shared interfaces*** breaks (Figure 12.9), 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 12.9 Flow of RHP packets when a link for shared interfaces brakes

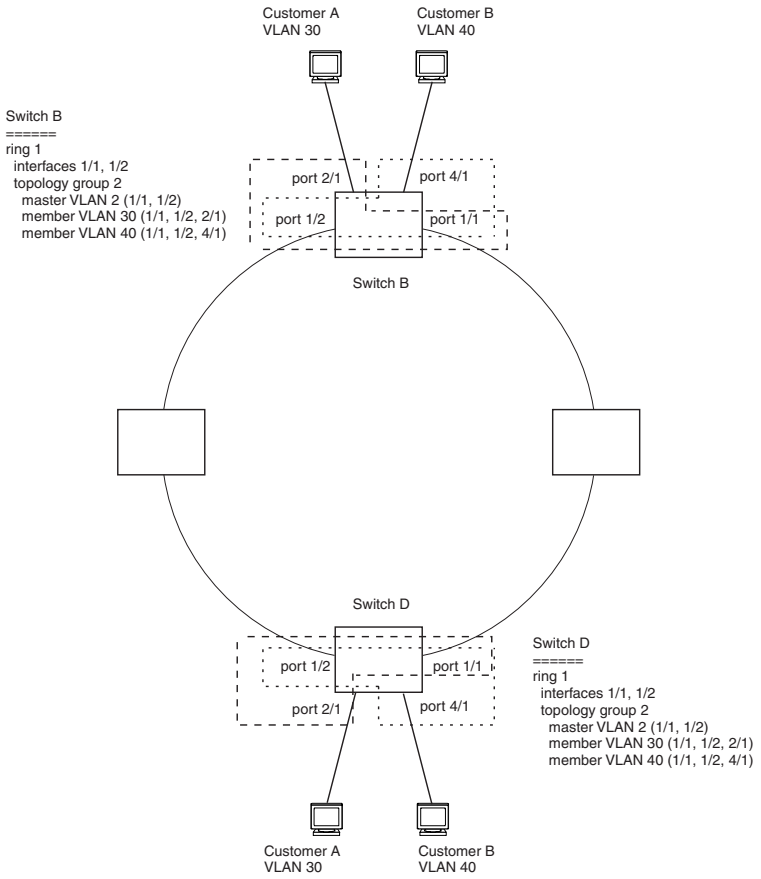


RHP packets follow this flow until the link is restored; then the RHP packet returns to it normal flow as shown in Figure 12.7.

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 12.10 shows an example.

Figure 12.10 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 12.10, 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, see "Topology Groups" on page 14-1.

See "MRP CLI Example" on page 12-16 for the configuration commands required to implement the MRP configuration shown in Figure 12.10.

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. See "Topology Groups" on page 14-1.
- Re-enable the interface you disabled to prevent a Layer 2 loop. Once MRP is enabled, MRP will prevent the Layer 2 loop.

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 BigIron RX) 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.

To configure MRP rings with shared interfaces, 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)# 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 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. See “Using MRP Diagnostics” .

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 12.1: 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, see “Configuring MRP” on page 12-11.

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>]

See “Displaying Topology Group Information” on page 14-3 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 12.2: 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

Table 12.2: CLI Display of MRP Ring Information (Continued)

This Field...	Displays...
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).
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.

Forwarding state	Whether MRP Forwarding is enabled on the interface. The forwarding state can be one of the following: • blocking – The interface is blocking Layer 2 data traffic and RHPs • disabled – The interface is down • forwarding – The interface is forwarding Layer 2 data traffic and RHPs • preforwarding – The interface is listening for RHPs but is blocking Layer 2 data traffic

Table 12.2: CLI Display of MRP Ring Information (Continued)

This Field...	Displays...
Active interface	The physical interfaces that are sending and receiving RHPs. **Note**: If a port is disabled, its state is shown as “disabled”. **Note**: If an interface is a trunk group, only the primary port of the group is listed.
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 interface state changes that have occurred. The state can be one of the states listed in the Forwarding state field.

MRP CLI Example

The following examples show the CLI commands required to implement the MRP configuration shown in Figure 12.10 on page 12-10.

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
```

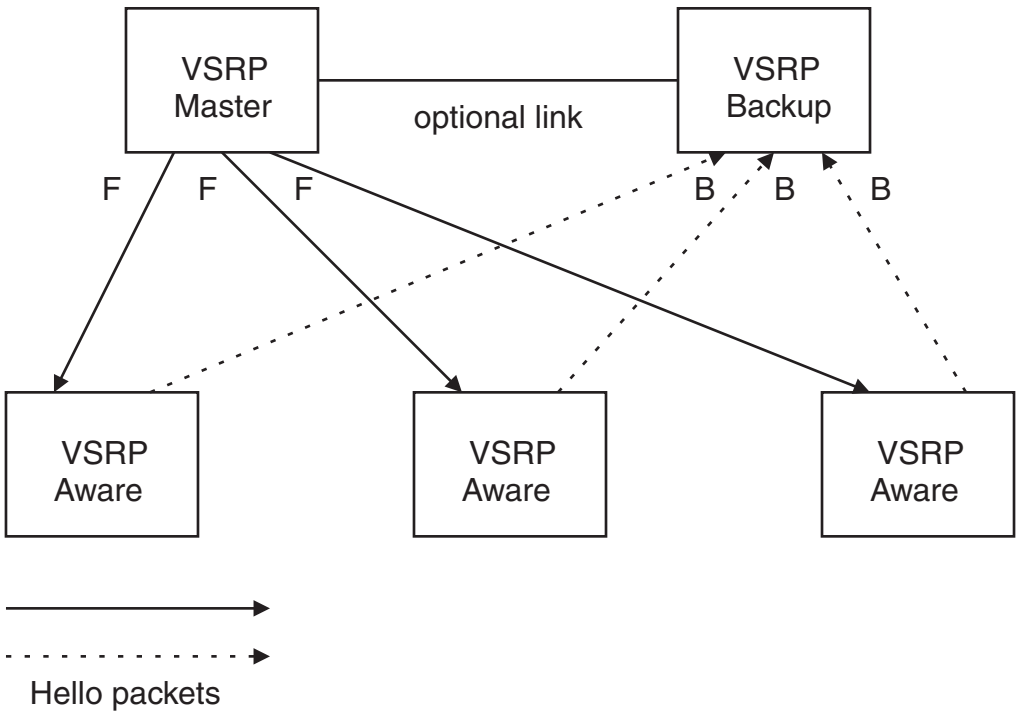

Chapter 13
Virtual Switch Redundancy Protocol (VSRP)

VSRP is a Foundry proprietary protocol that provides redundancy and sub-second failover in Layer 2 and Layer 3 mesh topologies. Based on the Foundry Virtual Router Redundancy Protocol Extended (VRRPE), VSRP provides one or more backups for the BigIron RX. If the active BigIron RX becomes unavailable, one of the backups takes over as the active device and continues forwarding traffic for the network.

You can use VSRP for the BigIron RX. Layer 2 and Layer 3 share the same VSRP configuration information.

Figure 13.1 shows a VSRP configuration.

Figure 13.1 VSRP mesh – redundant paths for Layer 2 and Layer 3 traffic



In this example, two BigIron RX devices are configured as redundant paths for VRID 1. On each BigIron RX, 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 Foundry 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 Foundry devices can use the redundant paths provided by the VSRP devices. In this example, three Foundry devices use the redundant paths. A Foundry device that is not itself configured for VSRP but is connected to a Foundry device that is configured for VSRP, is ***VSRP aware***. In this example, the three Foundry devices connected to the VSRP devices are VSRP aware. A Foundry 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 Foundry devices connected to the VSRP devices has a separate link to each of the VSRP devices.

Layer 2 Redundancy

VSRP provides Layer 2 redundancy. This means that Layer 2 links are backup up, but specific IP addresses are not backed up.

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:

- BigIron RX devices – The BigIron RX whose virtual routing interface has a higher IP address becomes the master.

VSRP Failover

Each Backup listens for Hello messages from the Master. The Hello messages indicate that the Master is still available. If the Backups stop receiving Hello messages from the Master, the election process occurs again and the Backup with the highest priority becomes the new Master.

Each Backup waits for a specific period of time, the Dead Interval, to receive a new Hello message from the Master. If the Backup does not receive a Hello message from the Master by the time the Dead Interval expires, the Backup sends a Hello message of its own, which includes the Backup's VSRP priority, to advertise the Backup's intent to become the Master. If there are multiple Backups for the VRID, each Backup sends a Hello message.

When a Backup sends a Hello message announcing its intent to become the Master, the Backup also starts a hold-down timer. During the hold-down time, the Backup listens for a Hello message with a higher priority than its own.

- If the Backup receives a Hello message with a higher priority than its own, the Backup resets its Dead Interval and returns to normal Backup status.
- If the Backup does not receive a Hello message with a higher priority than its own by the time the hold-down timer expires, the Backup becomes the new Master and starts forwarding Layer 2 traffic on all ports.

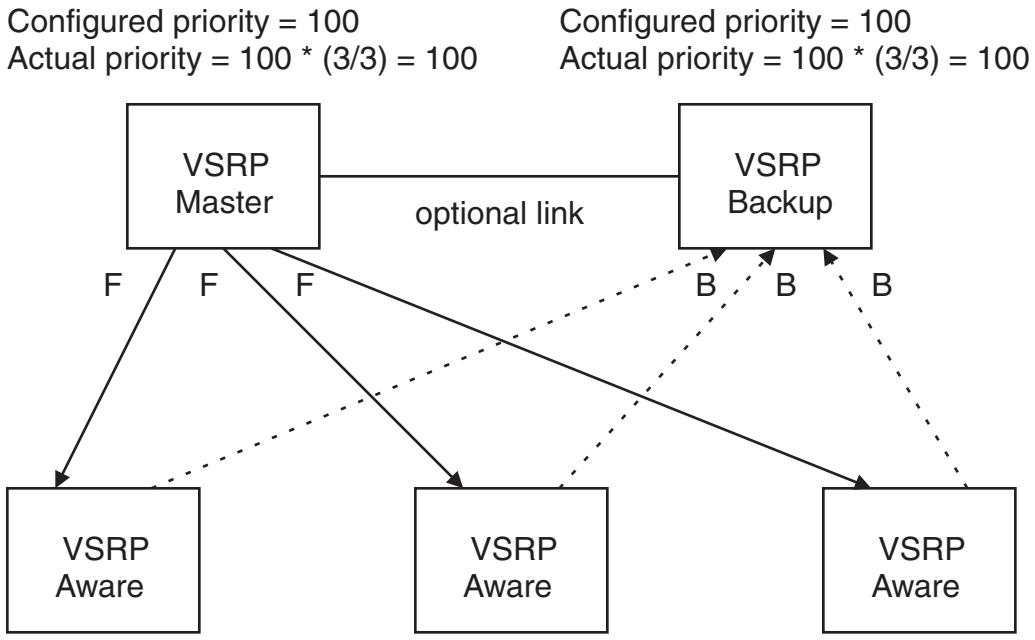
If you increase the timer scale value, each timer's value is divided by the scale value. To achieve sub-second failover times, you can change the scale to a value up to 10. This shortens all the VSRP timers to 10 percent of their configured values.

VSRP Priority Calculation

Each VSRP device has a VSRP priority for each VRID and its VLAN. The VRID is used during Master election for the VRID. By default, a device's VSRP priority is the value configured on the device (which is 100 by default). However, to ensure that a Backup with a high number of up ports for a given VRID is elected, the device reduces the priority if a port in the VRID's VLAN goes down. For example, if two Backups each have a configured priority

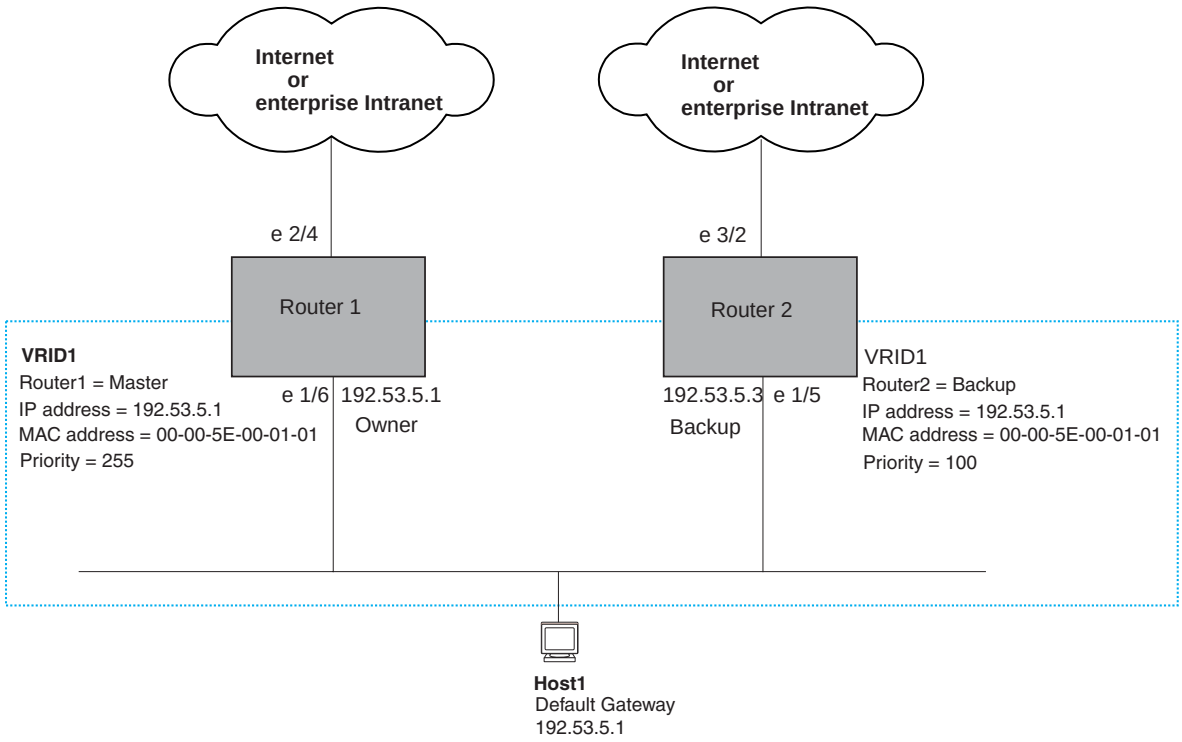
of 100, and have three ports in VRID 1 in VLAN 10, each Backup begins with an equal priority, 100. This is shown in Figure 13.2

Figure 13.2 VSRP priority



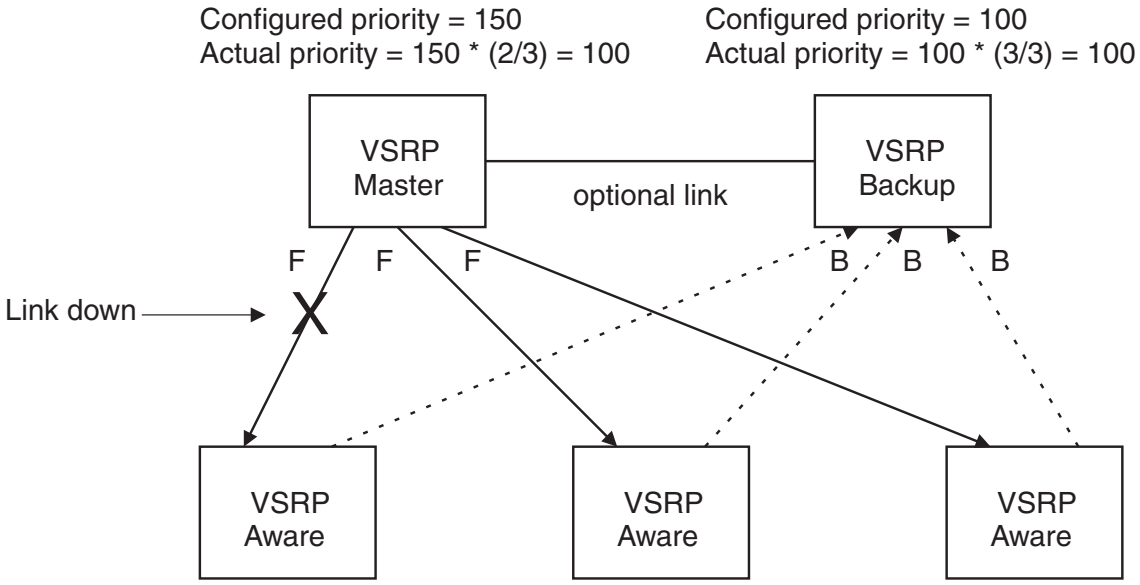
However, if one of the VRID's ports goes down on one of the Backups, that Backup's priority is reduced. If the Master's priority is reduced enough to make the priority lower than a Backup's priority, the VRID fails over to the Backup. Figure 13.3 shows an example.

Figure 13.3 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 13.3 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 13.4.

Figure 13.4 VSRP priority bias

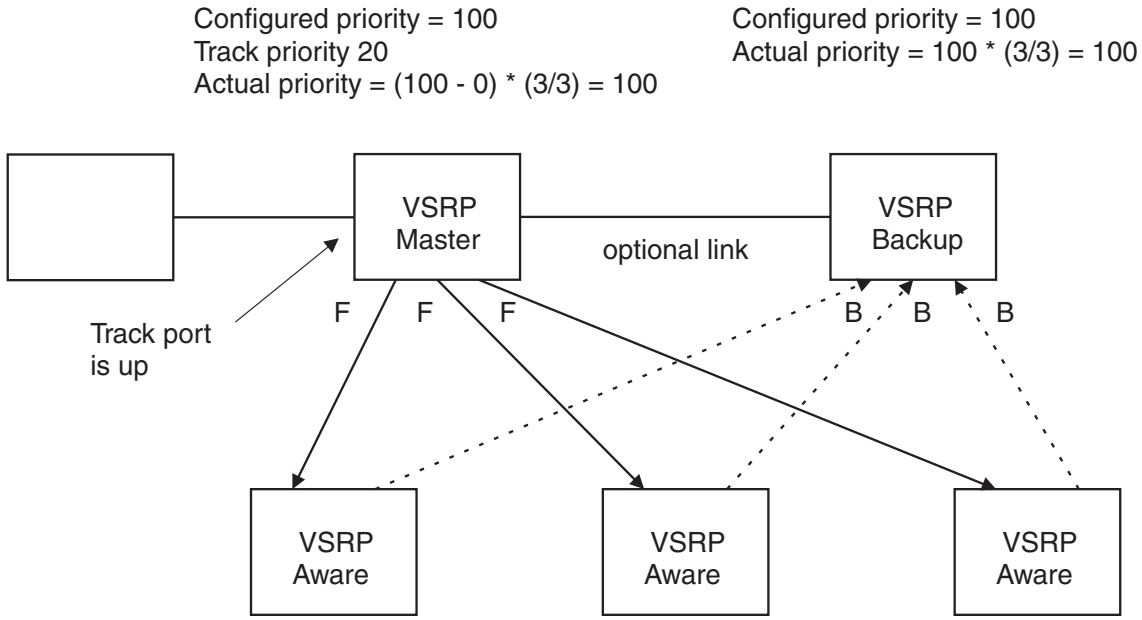


Track Ports

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.

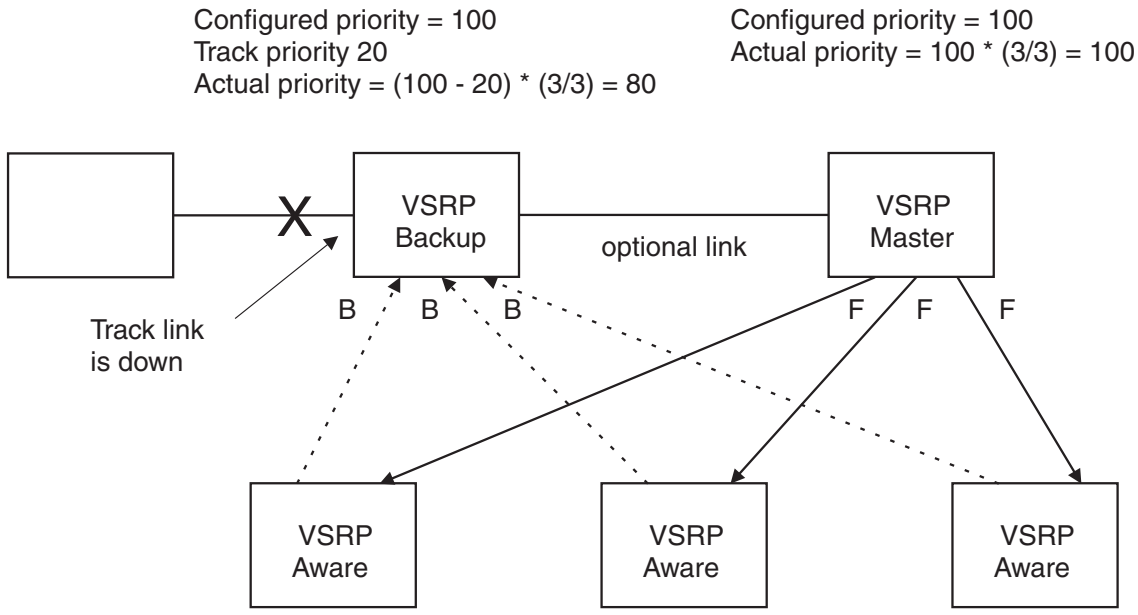
When you configure a track port, you assign a priority value to the port. If the port goes down, VSRP subtracts the track port's priority value from the configured VSRP priority. For example, if the you configure a track port with priority 20 and the configured VSRP priority is 100, the software subtracts 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 13.5 shows an example.

Figure 13.5 Track port priority



In Figure 13.5, 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 13.6.

Figure 13.6 Track port priority subtracted during priority calculation



MAC Address Failover on VSRP-Aware Devices

VSRP-aware devices maintain a record of each VRID and its VLAN. When the device has received a Hello message for a VRID in a given VLAN, the device creates a record for that VRID and VLAN and includes the port number in the record. Each subsequent time the device receives a Hello message for the same VRID and VLAN, the device checks the port number.

- If the port number is the same as the port that previously received a Hello message, the VSRP-aware device assumes that the message came from the same VSRP Master that sent the previous message.
- If the port number does not match, the VSRP-aware device assumes that a VSRP failover has occurred to a new Master, and moves the MAC addresses learned on the previous port to the new port.

The VRID records age out if unused. This can occur if the VSRP-aware device becomes disconnected from the Master. The VSRP-aware device will wait for a Hello message for the period of time equal to the following:

VRID Age = Dead Interval + Hold-down Interval + (3 x Hello Interval)

The values for these timers are determined by the VSRP device sending the Hello messages. If the Master uses the default timer values, the age time for VRID records on the VSRP-aware devices is as follows:

3 + 2 + (3 x 1) = 8 seconds

In this case, if the VSRP-aware device does not receive a new Hello message for a VRID in a given VLAN, on any port, the device assumes the connection to the Master is unavailable and removes the VRID record.

VSRP Parameters

Table 13.1 lists the VSRP parameters.

Table 13.1: VSRP Parameters

Parameter	Description	Default	See page...
Protocol	VSRP state **Note**: On a BigIron RX, you must disable VSRP to use VRRPE or VRRP.	Enabled	13-10
Virtual Router ID (VRID)	The ID of the virtual switch you are creating by configuring multiple devices as redundant links. You must configure the same VRID on each device that you want to use to back up the links.	None	13-9
Interface Parameters			
Authentication type	The type of authentication the VSRP devices use to validate VSRP packets. On a BigIron RX, 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.	No authentication	13-10
VRID Parameters			
VSRP device type	Whether the device is a VSRP Backup for the VRID. All VSRP devices for a given VRID are Backups.	Not configured	13-9
VSRP ports	The ports in the VRID's VLAN that you want to use as VRID interfaces. You can selectively exclude individual ports from VSRP while allowing them to remain in the VLAN.	All ports in the VRID's VLAN	13-10
VRID IP address	A gateway address you are backing up. Configuring an IP address provides VRRPE Layer 3 redundancy in addition to VSRP LAyer 2 redundancy. 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.	None	13-11

Table 13.1: VSRP Parameters (Continued)

Parameter	Description	Default	See page...
Backup priority	A numeric value that determines a Backup's preferability for becoming the Master for the VRID. During negotiation, the device with the highest priority becomes the Master. In VSRP, all devices are Backups and have the same priority by default. If two or more Backups are tied with the highest priority, the Backup with the highest IP address becomes the Master for the VRID.	100 for all Backups	13-11
Preference of timer source	When you save a Backup's configuration, the software can save the configured VSRP timer values or the VSRP timer values received from the Master. Saving the current timer values instead of the configured ones helps ensure consistent timer usage for all the VRID's devices. **Note**: The Backup always gets its timer scale value from the Master.	Configured timer values are saved	13-11
Time-to-Live (TTL)	The maximum number of hops a VSRP Hello packet can traverse before being dropped. You can specify from 1 – 255.	2	13-12
Hello interval	The amount of time between Hello messages from the Master to the Backups for a given VRID. The interval can be from 1 – 84 seconds.	One second	13-12
Dead interval	The amount of time 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	13-12
Backup Hello state and interval	The amount of time 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 sends Hello messages by default.	Disabled 60 seconds when enabled	13-13
Hold-down interval	The amount of time a Backup that has sent a Hello packet announcing its intent to become Master waits before beginning to forward traffic for the VRID. The hold-down interval prevents Layer 2 loops from occurring during VSRP's rapid failover. The interval can from 1 – 84 seconds.	2 seconds	13-13

Table 13.1: VSRP Parameters (Continued)

Parameter	Description	Default	See page...
Track priority	A VSRP priority value assigned to the tracked port(s). If a tracked port's link goes down, the VRID port's VSRP priority is reduced by the amount of the tracked port's priority.	5	13-13
Track port	A track port is a port or virtual routing interface that is outside the VRID but whose link state is tracked by the VRID. Typically, the tracked interface represents the other side of VRID traffic flow through the device. If the link for a tracked interface goes down, the VSRP priority of the VRID interface is changed, causing the devices to renegotiate for Master.	None	13-14
Backup preempt mode	Prevents a Backup with a higher VSRP priority from taking control of the VRID from another Backup that has a lower priority but has already assumed control of the VRID.	Enabled	13-14
VRID active state	The active state of the VSRP VRID.	Disabled	13-9
RIP Parameters			
Suppression of RIP advertisements	A BigIron RX that is running RIP normally advertises routes to a backed up VRID even when the BigIron RX is not currently the active BigIron RX for the VRID. Suppression of these advertisements helps ensure that other BigIron RX do not receive invalid route paths for the VRID.	Disabled (routes are advertised)	13-14

Configuring Basic VSRP Parameters

To configure VSRP, perform the following required tasks:

- Configure a port-based VLAN containing the ports for which you want to provide VSRP service.

  NOTE: If you already have a port-based VLAN but only want to use VSRP on a sub-set of the VLANs ports, you can selectively remove ports from VSRP service in the VLAN. See "Removing a Port from the VRID's VLAN" on page 13-10.

- Configure a VRID.
  - Specify that the device is a backup. Since VSRP, like VRRPE, does not have an "owner", all VSRP devices are backups. The active device for a VRID is elected based on the VRID priority, which is configurable.
  - Enable VSRP on the VRID.

The following example shows a simple VSRP configuration:

```
BigIron RX(config)# vlan 200
BigIron RX(config-vlan-200)# tag ethernet 1/1 to 1/8
BigIron RX(config-vlan-200)# vsrp vrid 1
BigIron RX(config-vlan-200-vrid-1)# backup
BigIron RX(config-vlan-200-vrid-1)# enable
```

Syntax: [no] vsrp vrid <num>

The <num> parameter specifies the VRID and can be from 1 – 255.

Syntax: [no] backup [priority <value>] [track-priority <value>]

This command is required. In VSRP, all devices on which a VRID are configured are Backups. The Master is then elected based on the VSRP priority of each device. There is no "owner" device as there is in VRRP.

For information about the command's optional parameters, see the following:

- "Changing the Backup Priority" on page 13-11
- "Changing the Default Track Priority" on page 13-13

Syntax: [no] enable | disable

Configuring Optional VSRP Parameters

The following sections describe how to configure optional VSRP parameters.

Disabling or Re-Enabling VSRP

VSRP is enabled by default on the BigIron RX. If you want to use VRRP or VRRPE for Layer 3 redundancy instead of VSRP, you disable VSRP first. To do so, enter the following command at the global CONFIG level:

```
BigIron RX(config)# no router vsrp
router vsrp is disabled. All vsrp config data will be lost when writing to flash
```

To re-enable the protocol, enter the following command:

```
BigIron RX(config)# router vsrp
```

Syntax: [no] router vsrp

Configuring Authentication

If the interfaces on which you configure the VRID use authentication, the VSRP packets on those interfaces also must use the same authentication. VSRP supports the following authentication types:

- No authentication – The interfaces do not use authentication. This is the default.
- Simple – The interfaces use a simple text-string as a password in packets sent on the interface. If the interfaces use simple password authentication, the VRID configured on the interfaces must use the same authentication type and the same password.

To configure a simple password, enter a command such as the following at the interface configuration level:

```
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.

Removing a Port from the VRID's VLAN

By default, all the ports in the VLAN on which you configure a VRID are interfaces for the VRID. You can remove a port from the VRID while allowing it to remain in the VLAN.

Removing a port is useful in the following cases:

- There is no risk of a loop occurring, such as when the port is attached directly to an end host.
- You plan to use a port in 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
```

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 you are configuring a BigIron RX for VSRP, you can specify an IP address to back up. When you specify an IP address, VSRP provides redundancy for the 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, see “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.

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>

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. See “VSRP Priority Calculation” on page 13-2 and “Changing the Default Track Priority” on page 13-13.

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, see “Changing the Default Track Priority” on page 13-13.

Saving the Timer Values Received from the Master

The Hello messages sent by a VRID’s master contain the VRID values for the following VSRP timers:

- Hello interval
- Dead interval
- Backup Hello interval
- Hold-down interval

By default, each Backup saves the configured timer values to its startup configuration file when you save the device's configuration.

You can configure a Backup to instead save the current timer values received from the Master when you save the configuration. Saving the current timer values instead of the configured ones helps ensure consistent timer usage for all the VRID's devices.

NOTE: The Backups always use the value of the timer scale received from the Master, regardless of whether the timer values that are saved in the configuration are the values configured on the Backup or the values received from the Master.

To configure a Backup to save the VSRP timer values received from the Master instead of the timer values configured on the Backup, enter the following command:

```
BigIron RX(config-vlan-200-vrid-1)# save-current-values
```

Syntax: [no] save-current-values

Changing the Time-To-Live (TTL)

A VSRP Hello packet's TTL specifies how many hops the packet can traverse before being dropped. A hop can be a 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 <num>

The <num> parameter specifies the interval and can be from 1 – 84 seconds. The default is 1 second.

NOTE: The default Dead interval is three times the Hello interval plus one-half second. Generally, if you change the Hello interval, you also should change the Dead interval on the Backups.

NOTE: If you change the timer scale, the change affects the actual number of seconds.

Changing the Dead Interval

The Dead interval is the number of seconds a Backup waits for a Hello message from the Master before determining that the Master is dead. The default is 3 seconds. This is three times the default Hello interval.

To change the Dead interval, enter a command such as the following at the configuration level for the VRID:

```
BigIron RX(config-vlan-200-vrid-1)# dead-interval 30
```

Syntax: [no] dead-interval <num>

The <num> parameter specifies the interval and can be from 1 – 84 seconds. The default is 3 seconds.

NOTE: If you change the timer scale, the change affects the actual number of seconds.

Changing the Backup Hello State and Interval

By default, Backups do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup to send Hello messages to the Master, enter a command such as the following at the configuration level for the VRID:

```
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 60 seconds by default. You can change the interval to be up to 3600 seconds.

To change the Backup Hello interval, enter a command such as the following at the configuration level for the VRID:

```
BigIron RX(config-vlan-200-vrid-1)# backup-hello-interval 180
```

Syntax: [no] backup-hello-interval <num>

The <num> parameter specifies the message interval and can be from 60 – 3600 seconds. The default is 60 seconds.

NOTE: If you change the timer scale, the change affects the actual number of seconds.

Changing the Hold-Down Interval

The hold-down interval prevents Layer 2 loops from occurring during failover, by delaying the new Master from forwarding traffic long enough to ensure that the failed Master is really unavailable.

To change the Hold-down interval, enter a command such as the following at the configuration level for the VRID:

```
BigIron RX(config-vlan-200-vrid-1)# hold-down-interval 4
```

Syntax: [no] hold-down-interval <num>

The <num> parameter specifies the hold-down interval and can be from 1 – 84 seconds. The default is 2 seconds.

NOTE: If you change the timer scale, the change affects the actual number of seconds.

Changing the Default Track Priority

When you configure a VRID to track the link state of other interfaces, if one of the tracked interface goes down, the software changes the VSRP priority of the VRID interface.

The software reduces the VRID priority by the amount of the priority of the tracked interface that went down. For example, if the VSRP interface's priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VSRP interface's priority to 40. If another tracked interface goes down, the software reduces the VRID's priority again, by the amount of the tracked interface's track priority.

The default track priority for all track ports is 1. You can change the default track priority or override the default for an individual track port.

- To change the default track priority, use the **backup track-priority** command, described below.
- To override the default track priority for a specific track port, use the **track-port** command. See "Specifying a Track Port" on page 13-14.

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. See "VSRP Priority Calculation" on page 13-2.

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

Suppressing RIP Advertisement from Backups

Normally, for Layer 3 a VSRP Backup includes route information for a backed up IP address in RIP advertisements. As a result, other BigIron RX devices 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.

NOTE: This parameter applies only if you specified an IP address to back up and is valid only on the BigIron RX.

To suppress RIP advertisements, enter the following commands:

```
Router2(config)# router rip
Router2(config-rip-router)# use-vrrp-path
```

Syntax: [no] use-vrrp-path

Displaying VSRP Information

You can display the following VSRP information:

- Configuration information and current parameter values for a VRID or VLAN
- The interfaces on a VSRP-aware device that are active for the VRID

Displaying VRID Information

To display VSRP information, enter the following command:

```
NetIron 40G# show vsrp vrid 10
VLAN 10
 auth-type no authentication
 VRID 10
 ========
  State        Administrative-status Advertise-backup Preempt-mode save-current
  standby      enabled               disabled         true         false

  Parameter        Configured Current    Unit/Formula
  priority         100        50         (100-0)*(1.0/2.0)
  hello-interval   1          1          sec/10
  dead-interval    3          3          sec/10
  hold-interval    3          3          sec/10
  initial-ttl      2          2          hops

  master router 219.130.154.186 expires in 00:00:00.5
  Member ports:      ethe 1/1 to 1/2
  Operational ports: ethe 1/1
  Forwarding ports:    None
```

On a devices where the VSRP Fast Start feature is enabled:

```
NetIron(config-vlan-100-vrid-100)#show vsrp vrid 100

VLAN 100
  auth-type no authentication
  VRID 100
  ========
   State      Administrative-status Advertise-backup Preempt-mode save-current
   master     enabled               disabled         true         false
   Parameter        Configured Current    Unit/Formula
   priority         100        50         (100-0)*(2.0/4.0)
   hello-interval   1          1          sec/1
   dead-interval    3          3          sec/1
   hold-interval    3          3          sec/1
   initial-ttl      2          2          hops

   next hello sent in 00:00:00.3
   Member ports:      ethe 2/5 to 2/8
   Operational ports: ethe 2/5 ethe 2/8
   Forwarding ports:  ethe 2/5 ethe 2/8
   Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

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, see "Displaying the Active Interfaces for a VRID" on page 13-18.

Table 13.2: CLI Display of VSRP VRID or VLAN Information

This Field...	Displays...
Total number of VSRP routers defined	The total number of VRIDs configured on this device.
VLAN	The VLAN on which VSRP is configured.
auth-type	The authentication type in effect on the ports in the VSRP VLAN.
VRID parameters	
VRID	The VRID for which the following information is displayed.
state	This device's VSRP state for the VRID. The state can be one of the following: • initialize – VSRP is not enabled on the VRID. If the state remains "initialize" after you enable VSRP on the VRID, make sure that the VRID is also configured on the other routers 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.

Table 13.2: CLI Display of VSRP VRID or VLAN Information (Continued)

This Field...	Displays...
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 seconds between Hello messages from the Master to the Backups for a given VRID.
dead-interval	The configured value for the dead interval. The dead interval is the number of seconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID. **Note**: If the value is 0, then you have not configured this parameter.
hold-interval	The number of seconds a Backup that intends to become the Master will wait before actually beginning to forward Layer 2 traffic for the VRID. If the Backup receives a Hello message with a higher priority than its own before the hold-down interval expires, the Backup remains in the Backup state and does not become the new Master.
initial-ttl	The number of hops a Hello message can traverse after leaving the device before the Hello message is dropped. **Note**: 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.

Table 13.2: CLI Display of VSRP VRID or VLAN Information (Continued)

This Field...	Displays...
master router	The IP address of the master router.
Member ports	The ports in the VRID.
Operational ports	The member ports that are currently up.
Forwarding ports	The member ports that are currently in the Forwarding state. Ports that are forwarding on the Master are listed. Ports on the Standby, which are in the Blocking state, are not listed.

Displaying the Active Interfaces for a VRID

On a VSRP-aware device, you can display VLAN and port information for the connections to the VSRP devices (Master and Backups).

To display the active VRID interfaces, enter the following command on the VSRP-aware device:

```
BigIron RX(config-vlan-200-vrid-1)# show vsrp aware

Aware port listing
VLAN ID  VRID  Last Port
100      1     3/2
200      2     4/1
```

Syntax: show vsrp aware

This display shows the following information when you use the **aware** parameter. For information about the display when you use the **vrid** <num> or **vlan** <vlan-id> parameter, see “Displaying VRID Information” on page 13-15.

Table 13.3: CLI Display of VSRP-Aware Information

This Field...	Displays...
VLAN ID	The VLAN that contains the VSRP-aware device’s connection with the VSRP Master and Backups.
VRID	The VRID.
Last Port	The most recent active port connection to the VRID. This is the port connected to the current Master. If a failover occurs, the VSRP-aware device changes the port to the port connected to the new Master. The VSRP-aware device uses this port to send and receive data through the backed up node.

VSRP Fast Start

It allows non-Foundry or non-VSRP aware devices that are connected to a Foundry device that is the VSRP Master to quickly switch over to the new Master when a VSRP failover occurs

This feature causes the port on a VSRP Master to restart when a VSRP failover occurs. When the port shuts down at the start of the restart, ports on the non-VSRP aware devices that are connected to the VSRP Master flush the MAC address they have learned for the VSRP master. After a specified time, the port on the previous VSRP Master (which now becomes the Backup) returns back online. Ports on the non-VSRP aware devices switch over to the new Master and learn its MAC address.

Special Considerations when Configuring VSRP Fast Start

- VSRP is sensitive to port status. When a port goes down, the VSRP instance lowers its priority based on the port up fraction. (see “VSRP Priority Calculation” on page 13-2 for more information on how priority is changed by port status). Since the VSRP fast start feature toggles port status by bringing ports down and up it can affect VSRP instances because their priorities get reduced when a port goes down. To avoid this, the VSRP fast start implementation keeps track of ports that it brings down and suppresses port down events for these ports (as concerns VSRP).
- Once a VSRP restart port is brought up by a VSRP instance, other VSRP instances (in Master state) that have this port as a member do not go to forwarding immediately. This is a safety measure that is required to prevent transitory loops. This could happen if a peer VSRP node gets completely cut off from this node and assumed Master state. In this case, where there are 2 VSRP instances that are in Master state and forwarding, the port comes up and starts forwarding immediately. This would cause a forwarding loop. To avoid this, the VSRP instance delays forwarding.

Recommendations for Configuring VSRP Fast Start

The following recommendations apply to configurations where multiple VSRP instances are running between peer devices sharing the same set of ports.

- Multiple VSRP instances configured on the same ports can cause VSRP instances to be completely cut off from peer VSRP instances. This can cause VSRP instances to toggle back and forth between master and backup mode. For this reason, we recommend that you configure VSRP fast start on a per port basis rather than for the entire VLAN.
- We recommend that VSRP peers have a directly connected port without VSRP fast start enabled on it. This allows protocol control packets to be received and sent even if other ports between the master and standby are down.
- The VSRP restart time should be configured based on the type of connecting device since some devices can take a long time to bring a port up or down (as long as several seconds). In order to ensure that the port restart is registered by neighboring device, the restart time may need to be changed to a value higher than the default value of 1 second.

Configuring VSRP Fast Start

The VSRP fast start feature can be enabled on a VSRP-configured Foundry 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:

```
BigIron RX(configure)# vlan 100
BigIron RX(configure-vlan-100)# vsrp vrid 1
BigIron RX(configure-vlan-100-vrid-1)# restart-ports 5
```

Syntax: [no] restart-ports <seconds>

This command shuts down all the ports that belong to the VLAN when a failover occurs. All the ports will have the specified VRID.

To configure a single port on a VSRP-configured device to shut down when a failover occurs, then restart after a period of time, enter the following command:

```
BigIron RX(configure)# interface ethernet 1/1
BigIron RX(configure-if-1/1)# vsrp restart-port 5
```

Syntax: [no] vsrp restart-port <seconds>

In both commands, the <seconds> parameter instructs the VSRP Master to shut down its port for the specified number of seconds before it starts back up. Enter a value between 1 – 120 seconds. The default is 1 second.

Displaying Ports that Have VSRP Fast Start Feature Enabled

The **show vsrp vrid** command shows the ports on which the VSRP fast start feature is enabled.

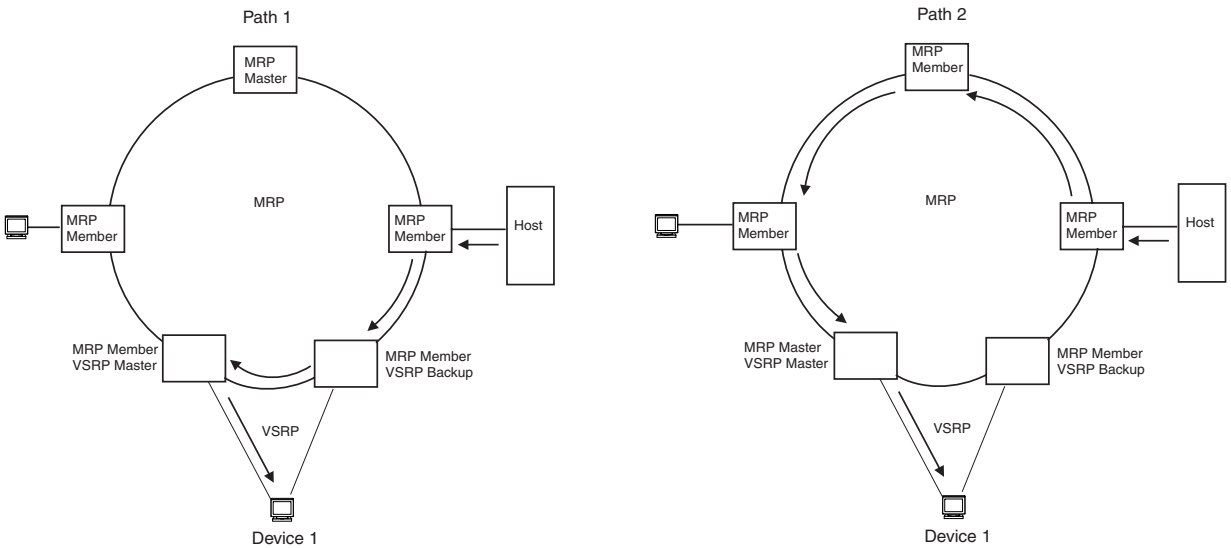
```
BigIron RX(config-vlan-100-vrid-100)#show vsrp vrid 100

VLAN 100
  auth-type no authentication
  VRID 100
  ========
   State      Administrative-status Advertise-backup Preempt-mode save-current
   master     enabled               disabled         true         false
   Parameter        Configured Current    Unit/Formula
   priority         100        50         (100-0)*(2.0/4.0)
   hello-interval   1          1          sec/1
   dead-interval    3          3          sec/1
   hold-interval    3          3          sec/1
   initial-ttl      2          2          hops

   next hello sent in 00:00:00.3
   Member ports:      ethe 2/5 to 2/8
   Operational ports: ethe 2/5 ethe 2/8
   Forwarding ports:  ethe 2/5 ethe 2/8
   Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

The "Restart ports:" line lists the ports that have the VSRP fast start enabled, and the downtime for each port. See Table 13.2 on page 13-16 to interpret the remaining information on the display.

VSRP and MRP Signaling

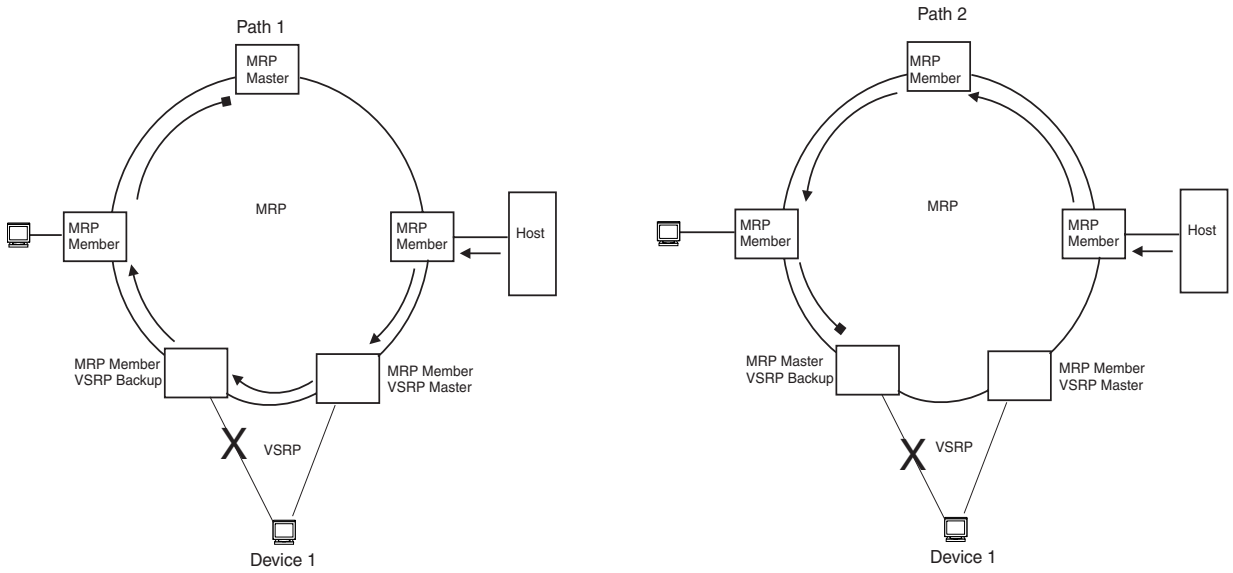
A device may connect to an MRP ring via VSRP to provide a redundant path between the device and the MRP ring. VSRP and MRP signaling, ensures rapid failover by flushing MAC addresses appropriately. The host on the MRP ring learns the MAC addresses of all devices on the MRP ring and VSRP link. From these 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 13.7 below shows two possible data paths from the host to Device 1.

Figure 13.7 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 13.8.

Figure 13.8 VSRP on MRP rings that failed over

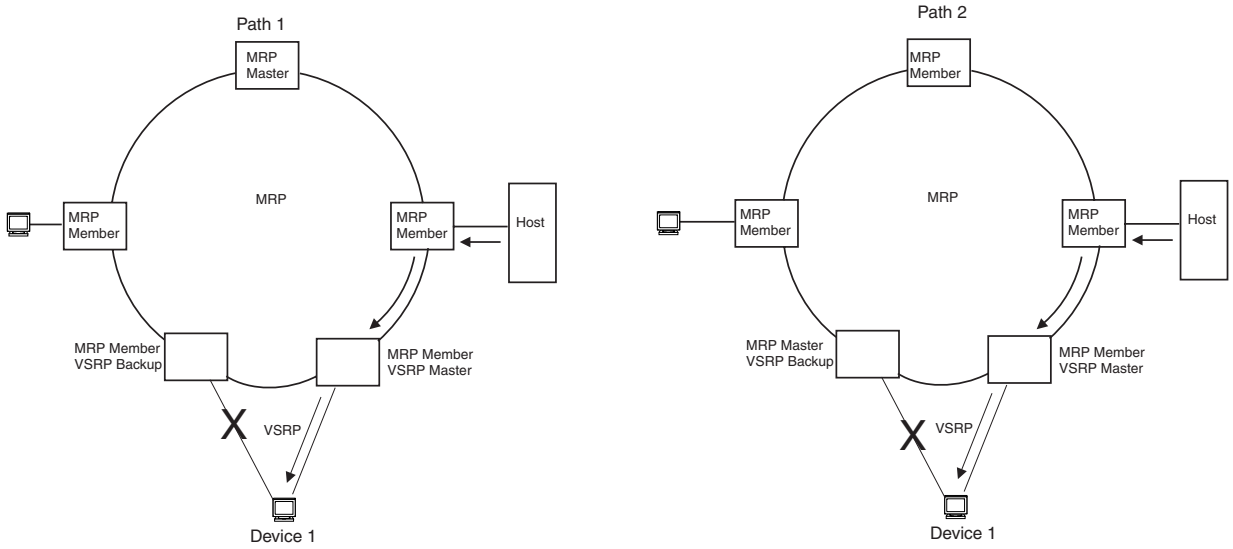


To ensure that MRP is informed of the topology change and to achieve convergence rapidly, a signaling process for the interaction between VSRP and MRP. When a VSRP node fails, a new VSRP master is selected. The new VSRP master finds all MRP instances impacted by the failover. Then each MRP instance does 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 13.9).

Figure 13.9 New path established



There are no CLI commands used to configure this process.

Chapter 14
Topology Groups

A topology group is a named set of VLANs that share a Layer 2 control protocol. Topology groups simplify configuration and enhance scalability of Layer 2 protocols by allowing you to run a single instance of a Layer 2 protocol on multiple VLANs. One instance of the Layer 2 protocol controls all the VLANs.

For example, if a BigIron RX 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 12-9.

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 next-highest numbered member VLAN as the new master VLAN. 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 3
================
 master-vlan 2
 member-vlan none

 Common control ports          L2 protocol
 ethernet 1/1                  MRP
 ethernet 1/2                  MRP
 ethernet 1/5                  VSRP
 ethernet 2/22                 VSRP
 Per vlan free ports
 ethernet 2/3                   Vlan 2
 ethernet 2/4                   Vlan 2
 ethernet 2/11                  Vlan 2
 ethernet 2/12                  Vlan 2
```

Syntax: show topology-group [<group-id>]

This display shows the following information.

Table 14.1: 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 15
Configuring VRRP and VRRPE

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 Foundry proprietary version of VRRP that overcomes limitations in the standard protocol. This protocol works only with Foundry devices.

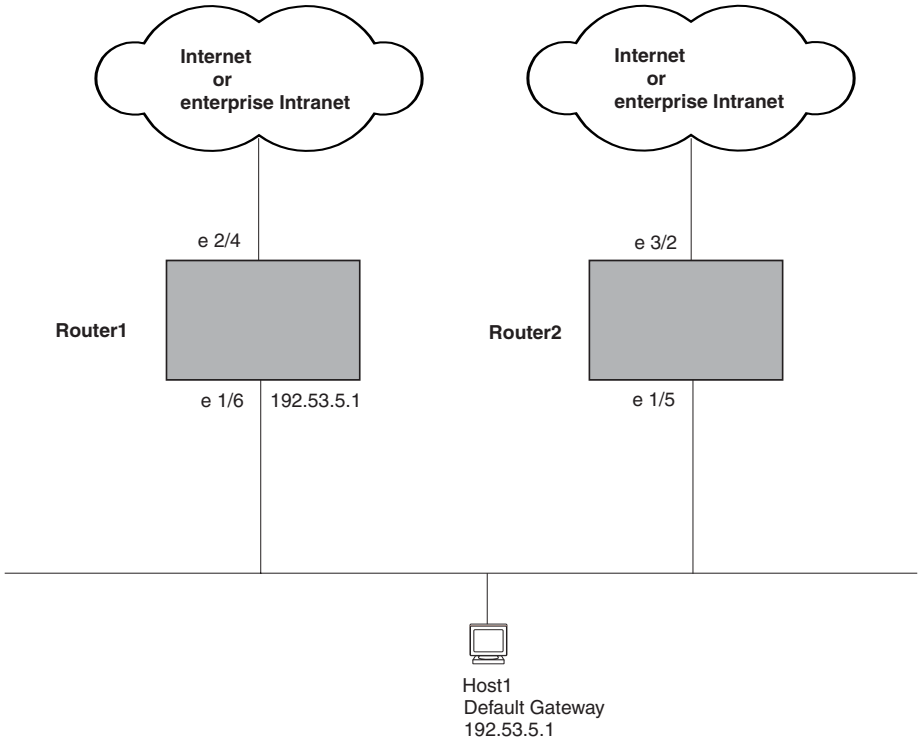
Overview of VRRP

This section presents the standard VRRP options and the options that Foundry 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 15.1.

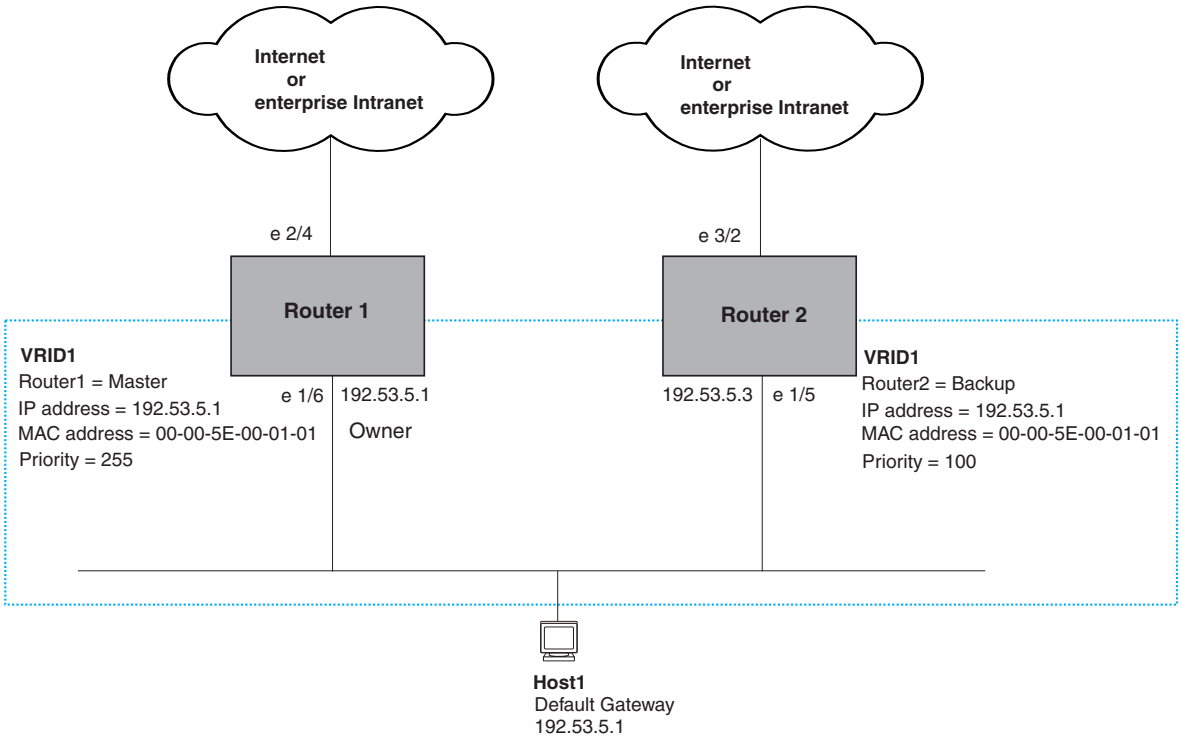
Figure 15.1 Router1 is Host1's default gateway but is a single point of failure



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 host(s). If VRRP is enabled as in Figure 15.2, Router 2 provides the default gateway out of the subnet if Router 1 fails.

Figure 15.2 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 15.2 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 15.2, 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 BigIron RX 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 15.2, 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).

Foundry's Enhancements of VRRP

Foundry 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

Foundry 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 15.2 on page 15-3, 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 15.2 on page 15-3, 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 15.2 on page 15-3, 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 Foundry 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 Foundry 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 Foundry'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 Backups
  - VRRP has an Owner and one or more Backups for each virtual router. The Owner is the router that has the IP address used for the virtual router. All the other routers supporting the virtual router are Backups.
  - 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 BigIron RX 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

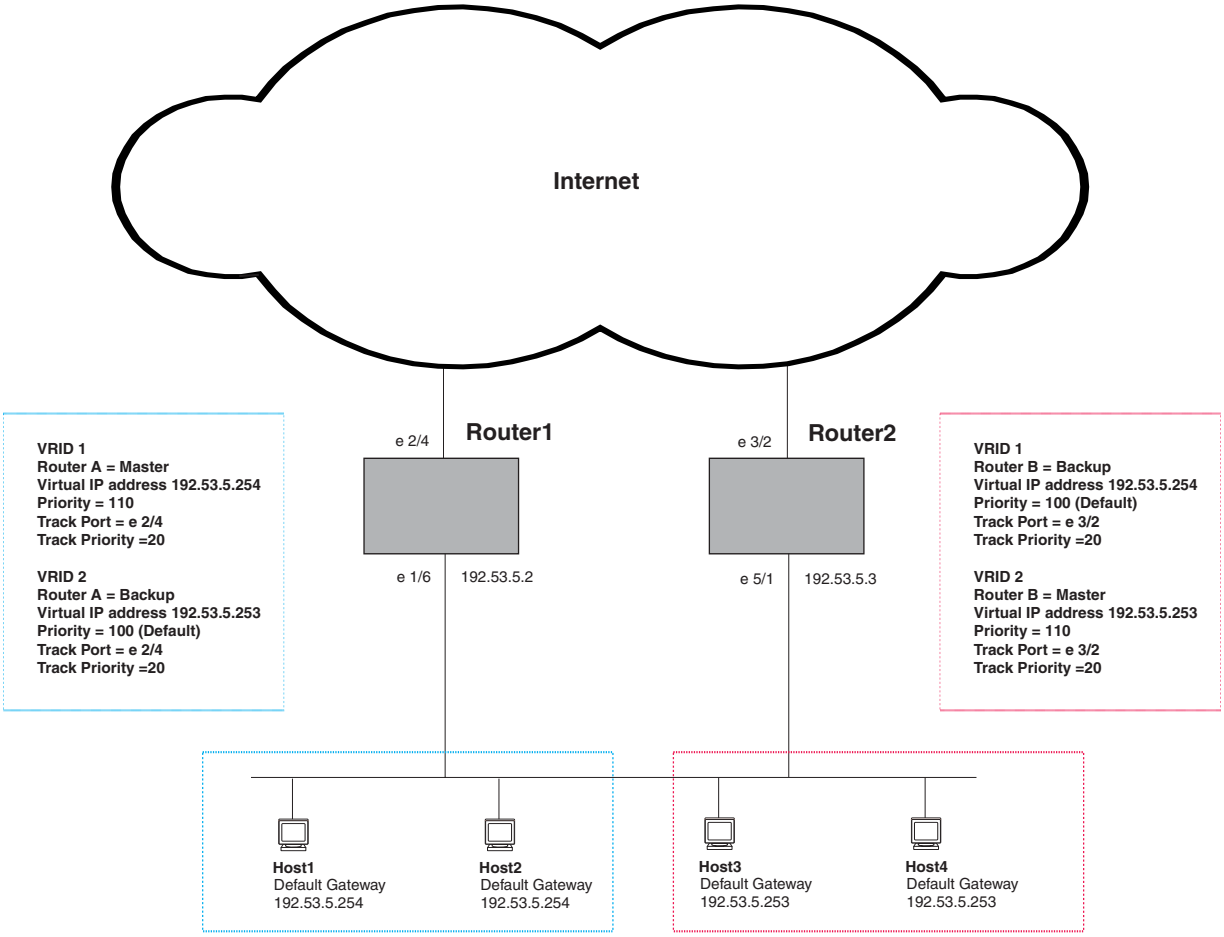
is 02-E0-52-<hash-value>-<vrid>, where <hash-value> is a two-octet hashed value for the IP address and <vrid> is the VRID.

- 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 15.3 shows an example of a VRRPE configuration.

Figure 15.3 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 BigIron RX configured for VRRPE can interoperate only with other BigIron RX.

VRRP and VRRPE Parameters

Table 15.1 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 15.1: VRRP and VRRPE Parameters

Parameter	Description	Default	See page...
Protocol	The Virtual Router Redundancy Protocol (VRRP) based on RFC 2338 or VRRP-Extended, Foundry's enhanced implementation of VRRP	Disabled **Note**: Only one of the protocols can be enabled at a time.	15-10 15-11
VRRP or VRRPE router	The BigIron RX's active participation as a VRRP or VRRPE router. Enabling the protocol does not activate the BigIron RX for VRRP or VRRPE. You must activate the BigIron RX as a VRRP or VRRPE router after you configure the VRRP or VRRPE parameters.	Inactive	15-10 15-11
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	15-10 15-11
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	15-10 15-11
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	15-4

Table 15.1: 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	15-5 15-13
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.	15-10 15-11
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	15-10 15-11
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	15-13
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	15-13

Table 15.1: VRRP and VRRPE Parameters (Continued)

Parameter	Description	Default	See page...
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	15-14
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	15-14
Track port	Another BigIron RX 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	15-4 15-14
Track priority	A VRRP or VRRPE priority value assigned to the tracked port(s). 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	15-4 15-15
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	15-15

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: 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.

The IP address you assign to the Owner must be an IP address configured on an interface that belongs to the virtual router.

See "Configuration Rules for VRRP" on page 15-11 for additional requirements.

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-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: 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.

See "Configuration Rules for VRRP" on page 15-11 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 Backup(s) for the virtual router.
- The Dead interval must be set to the same value on both the Owner and Backup(s) 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 Backup(s).

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 BigIron RX.

```
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>]

See the section "Authentication Type" on page 15-13 for information on the **auth-type no-auth** | **simple-text-auth** <auth-data> parameters.

Also, see "Configuration Rules for VRRPE" on page 15-12 additional information on how to configure VRRPE.

BigIron RX 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 BigIron RX devices.
- The Hello interval must be set to the same value on all the BigIron RX devices.
- The Dead interval must be set to the same value on all the BigIron RX devices.
- The track priority for a virtual router must be lower than the VRRPE priority.

NOTE: If you disable VRRPE, the BigIron RX 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

See "VRRP and VRRPE Parameters" on page 15-8 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. Foundry'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 BigIron RX. This capability is quite useful for tracking the state of the exit interface for the path for which the virtual router is providing redundancy. See “Track Ports and Track Priority” on page 15-4.

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. See "Track Port" on page 15-14.

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 BigIron RX is the Owner of the virtual router ("mode owner"), the BigIron RX'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 BigIron RX, enter the following command at any level of the CLI:

```
BigIron RX(config)# show ip vrrp-extended
Total number of VRRP-Extended routers defined: 4
Inte- VRID Current  P  State  Master IP       Backup IP       Virtual IP
rface      Priority           Address         Address         Address
--------------------------------------------------------------------------------
v10   1    100         Init   Unknown         Unknown         192.168.1.1
v20   1    100         Init   Unknown         Unknown         10.10.20.1
v30   1    100         Init   Unknown         Unknown         10.10.30.1
v100  1    100         Init   Unknown         Unknown         10.10.100.1
```

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. See “Displaying Detailed Information” on page 15-18.

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. See “Displaying Statistics” on page 15-21.

This display shows the following information.

Table 15.2: 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 BigIron RX. **Note**: The total applies only to the protocol the BigIron RX is running. For example, if the BigIron RX 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.
CurPri	The current VRRP or VRRPE priority of this BigIron RX 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.

Table 15.2: CLI Display of VRRP or VRRPE Summary Information (Continued)

This Field...	Displays...
State	This BigIron RX'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 BigIron RX is a Backup for the virtual router. • Master – This BigIron RX 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 brief

Total number of VRRP-Extended routers defined: 4

Interface v10
----------------
auth-type no authentication

VRID 1 (index 1)
 interface v10
 state initialize
 administrative-status enabled
 mode non-owner(backup)
 virtual mac 02e0.52e5.cd01
 priority 100
 current priority 100
 track-priority 5
 hello-interval 1 sec
 backup hello-interval 60 sec
 advertise backup disabled
 dead-interval 0 sec
 current dead-interval 0.0 sec
 preempt-mode false
 virtual ip address 192.168.1.1
```

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 summary information. See “Displaying Summary Information” on page 15-17.

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. See “Displaying Statistics” on page 15-21.

This display shows the following information.

Table 15.3: 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 BigIron RX. **Note**: The total applies only to the protocol the BigIron RX is running. For example, if the BigIron RX 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 BigIron RX’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 BigIron RX is a Backup for the virtual router. • master – This BigIron RX 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 15.3: CLI Display of VRRP or VRRPE Detailed Information (Continued)

This Field...	Displays...
mode	Indicates whether the BigIron RX 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. **Note**: This field applies only to VRRP. All BigIron RX 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 BigIron RX 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. See "Track Ports and Track Priority" on page 15-4.
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 BigIron RX 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. See "Hello Interval" on page 15-13.
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.

Table 15.3: CLI Display of VRRP or VRRPE Detailed Information (Continued)

This Field...	Displays...
current dead-interval	The current value of the dead interval. This is the value actually in use by this interface for the virtual router. **Note**: This field does not apply to VRRP Owners.
preempt-mode	Whether the backup preempt mode is enabled. **Note**: This field does not apply to VRRP Owners.
virtual ip address	The virtual IP addresses that this virtual router is backing up.
backup router <ip-addr> expires in <time>	The IP addresses of Backups that have advertised themselves to this Master by sending Hello messages. The <time> value indicates how long before the Backup expires. A Backup expires if you disable the advertise backup option on the Backup or the Backup becomes unavailable. Otherwise, the Backup's next Hello message arrives before the Backup expires. The Hello message resets the expiration timer. An expired Backup does not necessarily affect the Master. However, if you have not disabled the advertise backup option on the Backup, then the expiration may indicate a problem with the Backup. **Note**: This field applies only when Hello messages are enabled on the Backups (using the advertise backup option).
next hello sent in <time>	How long until the Backup sends its next Hello message. **Note**: This field applies only when this BigIron RX 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 BigIron RX itself will become the Master. **Note**: This field applies only when this BigIron RX 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:

```
BigIron RX#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:

```
BigIron RX(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 15.2 on page 15-3 and Figure 15.3 on page 15-7.

VRRP Example

To implement the VRRP configuration shown in Figure 15.2 on page 15-3, 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 15.2 on page 15-3 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 router(s) 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. See "Track Ports and Track Priority" on page 15-4.

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 15.3 on page 15-7, configure the VRRP Routers as shown in the following sections.

Configuring Router1

To configure VRRP Router1 in Figure 15.3 on page 15-7, 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 router(s) 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. See "Track Ports and Track Priority" on page 15-4.

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 16
Configuring Quality of Service

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.

This chapter describes how QoS is implemented and configured in the BigIron RX. The chapter contains the following sections:

- **Classification –** This section describes how the packets are classified and mapped into the forwarding queues by default. See “Classification” on page 16-1.
- **Marking –** This process allows you to change the 802.1p and DSCP information in a packet. See “Marking” on page 16-4.
- **Configuring ToS-Based QoS –** This section describes how to specify a trust level and enable marking. See “Configuring ToS-Based QoS” on page 16-6.
- **Changing QoS Mappings –** This section describes how to change the default priority mappings. See “Configuring the QoS Mappings” on page 16-7.
- **Determining Packet Drop Priority using WRED –** Weighted Random Early Detection (WRED) provides a mechanism for determining which packets to drop in a congested network. This section describes how WRED works. See “Determining Packet Drop Priority Using WRED” on page 16-11.
- **Configuring Packet Drop Priority using WRED –** This section describes how to configure Weighted Random Early Detection (WRED). See “Configuring Packet Drop Priority Using WRED” on page 16-13.
- **Scheduling Traffic for Forwarding –** The BigIron RX supports six different schemes for prioritizing traffic for forwarding in a congested network. This section describes each of these schemes and how to configure them. See “Scheduling Traffic for Forwarding” on page 16-15.
- **Multicast Traffic Engineering –** The BigIron RX supports limiting of multicast traffic from an individual packet processor. See “Configuring Multicast Traffic Engineering” on page 16-19.

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.

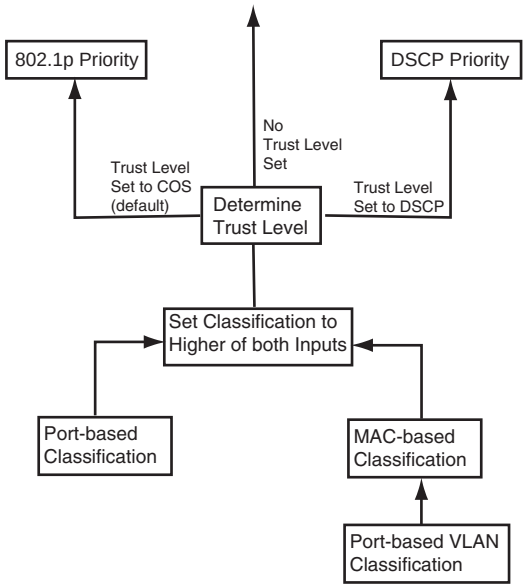
Packets on the BigIron RX are classified in up to eight traffic classes with values between 0 and 7. Packets with higher priority classifications are given a precedence for forwarding. These classes are determined by the following criteria in ascending order:

- **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 16.1.

Figure 16.1 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 BigIron RX. There are four queues designated as 0 to 3. The internal forwarding priority maps to one of these four queues as shown in Table 16.1 through Table 16.4. The mapping between the internal priority and the forwarding queue cannot be changed.

Table 16.1 through Table 16.4 show the default QoS mappings on the BigIron RX, which are used if the trust level for CoS or DSCP is enabled.

Table 16.1: 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 16.2: 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 16.3: 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 16.4: 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 16.1 through Table 16.4. This mapping is used for DSCP marking when trust level is COS. See “Changing the CoS –> DSCP Mappings” on page 16-7.
- **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. See “Changing the DSCP –> DSCP Mappings” on page 16-7.
- **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 16.1 through Table 16.4. This mapping is used for COS marking and determining the internal priority when the trust level is DSCP. See “Changing the DSCP –> Internal Forwarding Priority Mappings” on page 16-8.
- **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 16.1 through Table 16.4. 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 16-8.

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 16-2.

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 Foundry 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 BigIron RX 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 and/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 Foundry device maps a packet's 802.1p or DSCP value to an internal forwarding priority. The default mappings are listed in Table 16.1 through Table 16.4. 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, see “Displaying QoS Configuration Information” on page 16-10.

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, see “Displaying QoS Configuration Information” on page 16-10.

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 16.5: 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 BigIron RX 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:

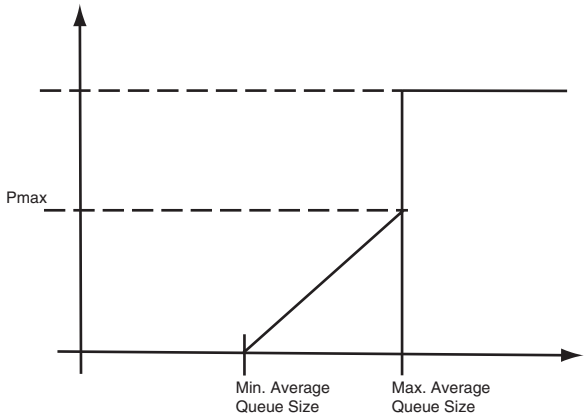
- **Statistical Average-Q-Size** – The statistical average size of the queue calculated over time on the switch.
- **Current-Q-Size** – The current size of the queue as calculated on the switch.
- **Wq** – This variable specifies the weights that should be given to the **current queue size** and the **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 17 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 - (see “Calculating avg-q-size” on page 16-12). 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 16-13.

Figure 17 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 W**q** 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.

$$\text{Pdrop} = \frac{\text{pkt-size}}{\text{pkt-size-max}} * \text{Pmax} * \frac{\text{(avg-q-size - min-avg-q size)}}{\text{(max-avg-q-size - min-avg-q size)}}$$

Using WRED with Rate Limiting

When rate limiting is configured on a BigIron RX, 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, see “Determining Packet Drop Priority Using WRED” on page 16-11. This section describes how to configure the parameters described in that section to enable the use of WRED on a BigIron RX.

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 BigIron RX. 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 described in “Determining Packet Drop Priority Using WRED” on page 16-11 is configured as the **averaging-weight** parameter. To set the wq parameter to 32%, use the following command:

```
BigIron RX(config)#qos queue-type 1 wred averaging-weight 32
```

This gives the current queue size a weight of 32% over the statistical average queue size.

Syntax: [no] qos queue-type <*queue-number*> wred averaging-weight <avg-weight>

The <*queue-number*> variable is the number of the forwarding queue that you want to configure the **averaging-weight** (Wq) parameter for. There are four forwarding queues on BigIron RX. They are numbered 0 to 3 with zero as the lowest priority queue and three the highest.

The <*avg-weight*> variable is the weight-ratio between instantaneous and average queue sizes. It is described as the Wq parameter in “Determining Packet Drop Priority Using WRED” on page 16-11. It is expressed as a percentage. The default value is 100%.

Configuring the Drop Precedence Parameters

A single instance of WRED being used for determining packet drop priority is described in "Determining Packet Drop Priority Using WRED" on page 16-11. In practice, an instance must be configured for each or the four forwarding queues that you want it to be active in. In addition, if you are using WRED drop precedence parameters with rate limiting, WRED can be configured to apply an different instance based on whether the traffic is less or greater than the maximum burst rate.

This section describes how to set the following drop precedence parameters:

- "Setting the Maximum Drop Probability"
- "Setting the Minimum and Maximum Average Queue Size"
- "Setting the Maximum Packet Size"

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 BigIron RX. 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 that you want to configure drop-precedence for. This value can be either 0 or 2 as described:

2 – If the traffic is policed using rate limiting and its rate exceeds the **maximum burst rate**.

0 – If the traffic is policed using rate limiting and its rate is below the **maximum burst rate**.

NOTE: While values 0 - 3 are allowed to be entered, only values 0 and 2 can be used.

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.

Setting the Minimum and Maximum Average Queue Size

To set the maximum average queue size 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-number> wred drop-precedence <policing-status> 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-number> wred drop-precedence <policing-status> min-avg-queue-size <min-size>

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 the BigIron RX. 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 that you want to configure drop-precedence for. This value can be either 0 or 2 as described:

2 – If the traffic is policed using rate limiting and its rate exceeds the **maximum burst rate**.

0 – If the traffic is policed using rate limiting and its rate is below the **maximum burst rate**.

NOTE: While values 0 - 3 are allowed to be entered, only values 0 and 2 can be used.

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 value is 32768 Kbytes.

The *<max-size>* variable is the average queue size above which all packets are dropped. (1 - 32768) (KBytes) in multiples of 64K. The default value is 32768 Kbytes.

Setting the Maximum Packet Size

To set the maximum packet size to 16 bytes, use the following command:

```
BigIron RX(config)#qos queue-type 1 wred drop-precedence 0 packet-size-max 16
```

Syntax: [no] qos queue-type <queue-number> wred drop-precedence <policing-status> packet-size-max <pkt-size>

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 the BigIron RX. 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 that you want to configure drop-precedence for. This value can be either 0 or 1 as described:

2 – If the traffic is policed using rate limiting and its rate exceeds the **maximum burst rate**.

0 – If the traffic is policed using rate limiting and its rate is below the **maximum burst rate**.

NOTE: While values 0 - 3 are allowed to be entered, only values 0 and 2 can be used.

The *<pkt-size>* variable is the pkt-size-max variable used in the equation described in “Calculating Packets That Are Dropped” on page 16-13. The default value is 512.

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 BigIron RX 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 16-11 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 isn't 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"

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:

$$\text{Weight of q (x)} = \frac{\text{q (x)}}{\text{q0 + q1 + q2 +q3}}$$

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**:

$$\text{Weight of q3} = \frac{\text{20}}{\text{5 + 10 + 15 + 20}}$$

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
BigIron RX(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.

See "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:

```
BigIron RX(config)# interface ethernet 1/1
BigIron RX(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.

See "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.

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.

Chapter 18
Configuring IP

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

Basic configuration consists of adding IP addresses and enabling a route exchange protocol. See “Configuring IP Addresses” on page 18-10.

To change some of the IP parameters from their default values or to view configuration information or statistics, see the following:

The IP Packet Flow

Figure 18.1 shows how an IP packet moves through a BigIron RX.

Figure 18.1 IP Packet flow through a BigIron RX

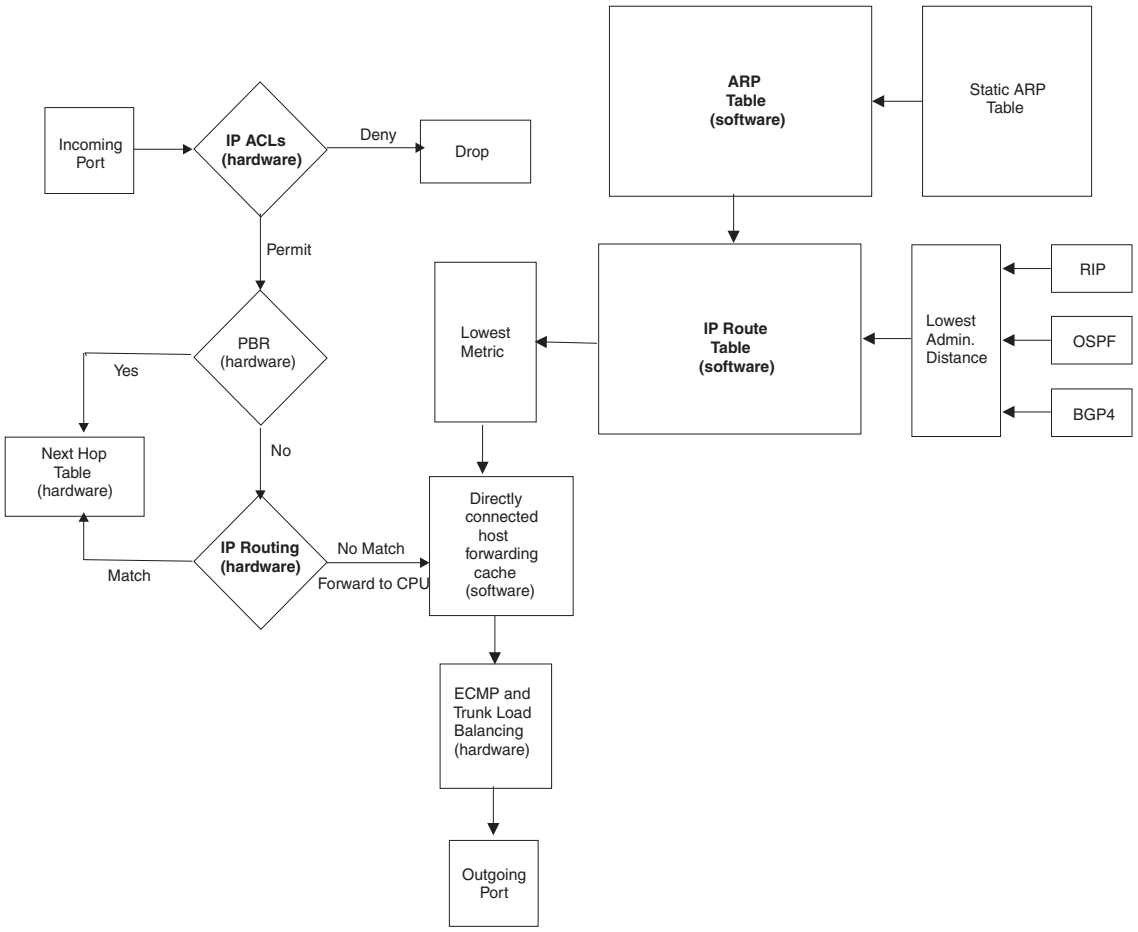


Figure 18.1 shows the following packet flow:

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

You also can change the capacity of the following tables by changing the memory allocation for the table:

ARP Cache Table

The Address Resolution Protocol (ARP) is supported on the BigIron RX. See "Configuring ARP Parameters" on page 18-23.

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

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

The software can learn an entry when the BigIron RX 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 BigIron RX 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 BigIron RX.

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, see the following:

- "Displaying the ARP Cache" on page 18-50
- "Displaying the Static ARP Table" on page 18-51

To configure other ARP parameters, see "Configuring ARP Parameters" on page 18-23.

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

- For dynamic entries, see the "Displaying and Modifying System Parameter Default Settings" on page 4-13. The ip-arp parameter controls the ARP cache size.
- For static entries, see "Changing the Maximum Number of Entries the Static ARP Table Can Hold" on page 18-26. 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, see "Displaying the IP Route Table" on page 18-54.

To configure a static IP route, see "Configuring Static Routes" on page 18-30.

To clear a route from the IP route table, see "Clearing IP Routes" on page 18-56.

To increase the size of the IP route table for learned and static routes, see "Displaying and Modifying System Parameter Default Settings" on page 4-13.

- For learned routes, modify the ip-route parameter.
- For static routes, modify the ip-static-route parameter.

IP Forwarding Cache

The BigIron RX 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 BigIron RX 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, see “Displaying the Forwarding Cache” on page 18-52.

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 18.1 lists the IP global parameters for the BigIron RX, their default values, and where to find configuration information.

Table 18.1: 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.	18-10

Table 18.1: IP Global Parameters (Continued)

Parameter	Description	Default	See page...
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.	18-21
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	18-19
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	18-23
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	18-24
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. See Table 18.2 on page 18-9.	Ten minutes	18-24
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	18-25
Static ARP entries	An ARP entry you place in the static ARP table. Static entries do not age out.	No entries	18-25
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	18-26

Table 18.1: IP Global Parameters (Continued)

Parameter	Description	Default	See page...
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. See Table 18.2 on page 18-9.	Disabled	18-26
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.	18-27
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	18-27
Internet Control Message Protocol (ICMP) messages	The BigIron RX 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. See Table 18.2 on page 18-9.	Enabled	18-28 18-29
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. See Table 18.2 on page 18-9.	Disabled	18-40
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	18-45
Maximum Frame Size	You can set a maximum frame size of IP packets that are forwarded on all ports of a PPCR.		18-18

Table 18.1: IP Global Parameters (Continued)

Parameter	Description	Default	See page...
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	18-13
DNS default gateway addresses	A list of gateways attached to the router through which clients attached to the router can reach DNSs.	None configured	18-13
IP load sharing	A Foundry 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	18-38
Maximum IP load sharing paths	The maximum number of equal-cost paths across which the BigIron RX is allowed to distribute traffic.	Four	18-38
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	25-6 26-27 27-20
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	18-36
Static route	An IP route you place in the IP route table.	No entries	18-30
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.	18-21

IP Interface Parameters

Table 18.2 lists the interface-level IP parameters for the BigIron RX, their default values, and where to find configuration information.

Table 18.2: 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 BigIron RX has separate IP addresses on individual interfaces.	None configured[a]	18-10
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	18-18
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	18-19
ARP age	Locally overrides the global setting. See Table 18.1 on page 18-5.	Ten minutes	18-24
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	25-4
Directed broadcast forwarding	Locally overrides the global setting. See Table 18.1 on page 18-5.	Disabled	18-26
ICMP Router Discovery Protocol (IRDP)	Locally overrides the global IRDP settings. See Table 18.1 on page 18-5.	Disabled	18-41
ICMP Redirect messages	Locally overrides the global setting. See Table 18.1 on page 18-5.	Enabled	18-29

Table 18.2: IP Interface Parameters (Continued)

Parameter	Description	Default	See page...
DHCP gateway stamp	The router can assist DHCP/BootP Discovery packets from one subnet to reach DHCP/BootP servers on a different subnet by placing the IP address of the router interface that receives the request in the request packet's Gateway field. You can override the default and specify the IP address to use for the Gateway field in the packets. **Note**: UDP broadcast forwarding for client DHCP/BootP requests (bootpc) must be enabled and you must configure an IP helper address (the server's IP address or a directed broadcast to the server's subnet) on the port connected to the client.	The lowest-numbered IP address on the interface that receives the request	18-45
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	18-43
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	18-44

a.Some devices have a factory default, such as 209.157.22.154, used for troubleshooting during installation. For the BigIron RX, 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 BigIron RX 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 BigIron RX 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. See "Changing the Network Mask Display to Prefix Format" on page 18-13.

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 BigIron RX 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 BigIron RX and other devices.

You can configure up to eight loopback interfaces on a BigIron RX.

You can add up to 24 IP addresses to each loopback interface.

NOTE: If you configure the BigIron RX 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 BigIron RX. See “Adding a Loopback Interface” on page 27-40 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, see “Assigning an IP Address to an Ethernet Port” on page 18-11.

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 BigIron RX.

NOTE: Other sections in this chapter that describe how to configure interface parameters also apply to virtual interfaces.

NOTE: The BigIron RX 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, see “Assigning an IP Address to an Ethernet Port” on page 18-11.

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). 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 BigIron RX using Telnet or Secure Shell (SSH) CLI connections or the Web management interface, you must configure an IP address for the BigIron RX. Optionally, you also can specify the default gateway.

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>

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
```

To specify the BigIron RX's default gateway, enter a command such as the following:

```
BigIron RX(config)# ip default-gateway 192.45.6.1 255.255.255.0
```

Syntax: ip default-gateway <ip-addr>

or

Syntax: ip default-gateway <ip-addr>/<mask-bits>

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 BigIron RX and thereby recognize all hosts within that domain. After you define a domain name, the BigIron RX 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 BigIron RX 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 BigIron RX 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>

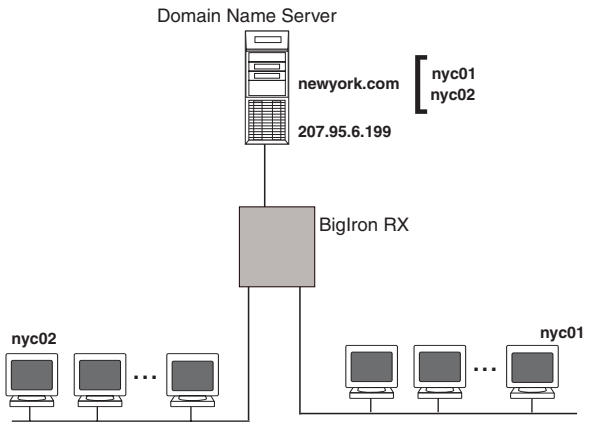
Syntax: ip dns server-address <ip-addr> [<ip-addr>] [<ip-addr>] [<ip-addr>]

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.

Using a DNS Name To Initiate a Trace Route

Suppose you want to trace the route from a BigIron RX to a remote server identified as NYC02 on domain newyork.com.

Figure 18.2 Querying a host on the newyork.com domain



Because the newyork.com domain is already defined on the BigIron RX, 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.

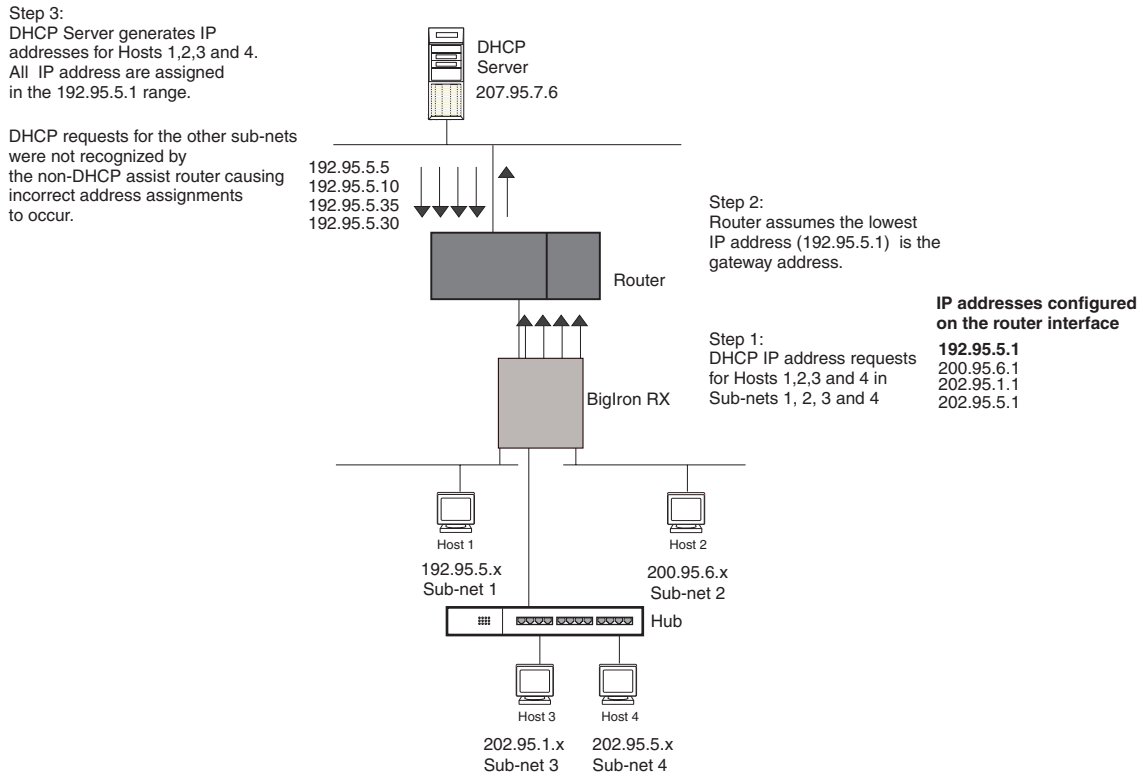
Configuring DHCP Assist

DHCP Assist allows a BigIron RX to assist a router that is performing multi-netting on its interfaces as part of its DHCP relay function.

DHCP Assist ensures that a DHCP server that manages multiple IP subnets can readily recognize the requester's IP subnet, even when that server is not on the client's local LAN segment. The BigIron RX does so by stamping each request with its IP gateway address in the DHCP discovery packet.

By allowing multiple subnet DHCP requests to be sent on the same wire, you can reduce the number of router ports required to support secondary addressing as well as reduce the number of DHCP servers required, by allowing a server to manage multiple subnet address assignments.

Figure 18.3 DHCP requests in a network without DHCP Assist on the BigIron RX



In a network operating without DHCP Assist, hosts can be assigned IP addresses from the wrong subnet range because a router with multiple subnets configured on an interface cannot distinguish among DHCP discovery packets received from different subnets.

For example, in Figure 18.3 a host from each of the four subnets supported on a BigIron RX requests an IP address from the DHCP server. These requests are sent transparently to the router. Because the router is unable to determine the origin of each packet by subnet, it assumes the lowest IP address or the 'primary address' is the gateway for all ports on the BigIron RX and stamps the request with that address.

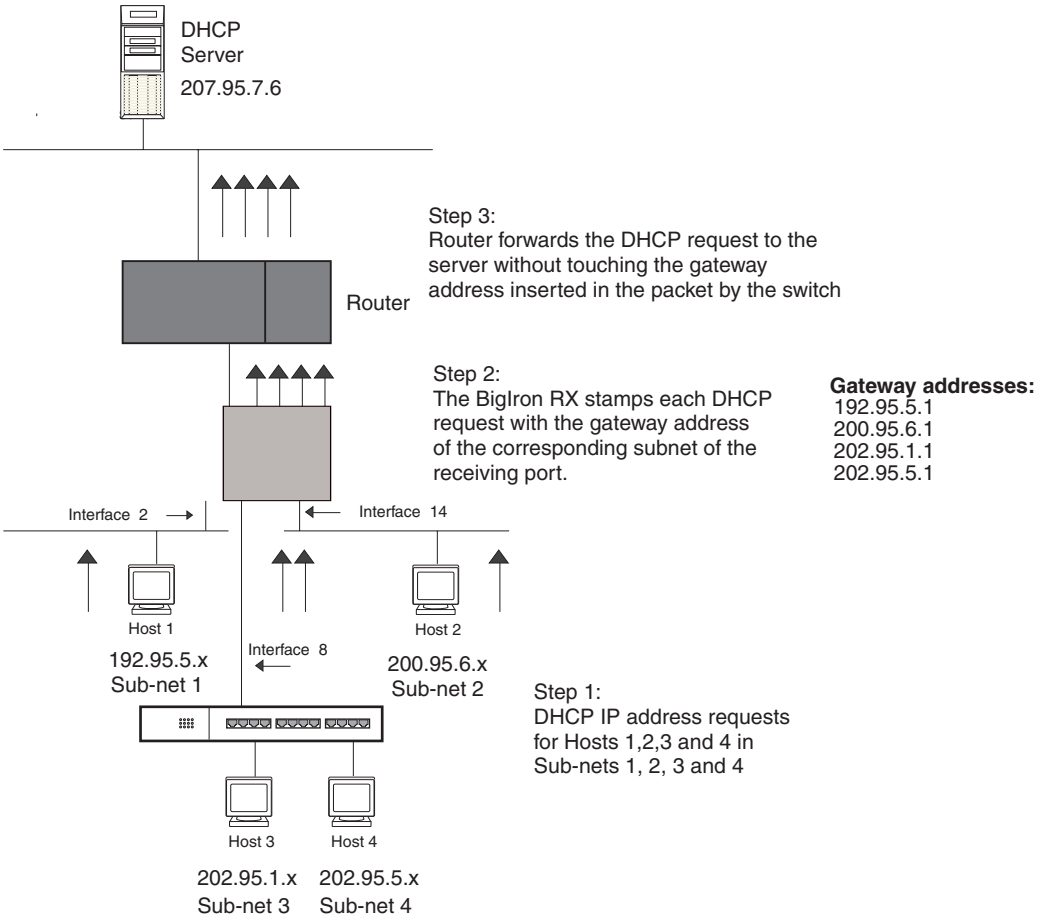
When the DHCP request is received at the server, it assigns all IP addresses within that range only.

With DHCP Assist enabled on a BigIron RX, correct assignments are made because the BigIron RX provides the stamping service.

How DHCP Assist Works

Upon initiation of a DHCP session, the client sends out a DHCP discovery packet for an address from the DHCP server as seen in Figure 18.4. When the DHCP discovery packet is received at a BigIron RX with the DHCP Assist feature enabled, the gateway address configured on the receiving interface is inserted into the packet. This address insertion is also referred to as stamping.

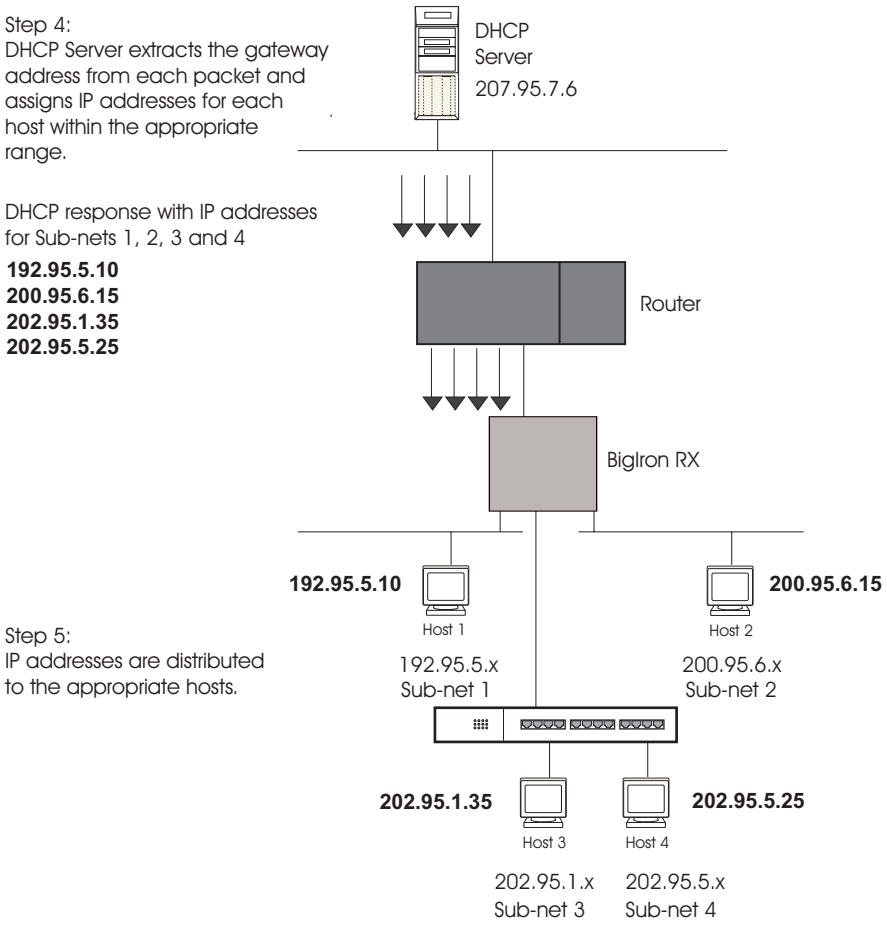
Figure 18.4 DHCP requests in a network with DHCP Assist



When the stamped DHCP discovery packet is then received at the router, it is forwarded to the DHCP server. The DHCP server then extracts the gateway address from each request and assigns an available IP address within the corresponding IP subnet (Figure 18.5). The IP address is then forwarded back to the workstation that originated the request.

NOTE: The DHCP relay function of the connecting router needs to be turned on.

Figure 18.5 DHCP offers are forwarded back toward the requestors



Configuring DHCP Gateway List

You can associate a gateway list with a port. You must configure a gateway list when DHCP Assist is enabled on a BigIron RX. The gateway list contains a gateway address for each subnet that will be requesting addresses from a DHCP server. The list allows the stamping process to occur. Each gateway address defined on the BigIron RX corresponds to an IP address of the Foundry router interface or other router involved.

Up to eight addresses can be defined for each gateway list in support of ports that are multi-homed. When multiple IP addresses are configured for a gateway list, the BigIron RX inserts the addresses into the discovery packet in a round robin fashion.

Up to 32 gateway lists can be defined for each BigIron RX.

EXAMPLE:

To create the configuration indicated in Figure 18.4 and Figure 18.5:

```
BigIron RX(config)# dhcp-gateway-list 1 192.95.5.1
BigIron RX(config)# dhcp-gateway-list 2 200.95.6.1
BigIron RX(config)# dhcp-gateway-list 3 202.95.1.1 202.95.5.1
BigIron RX(config)# int e 2
BigIron RX(config-if-e10000-2)# dhcp-gateway-list 1
BigIron RX(config-if-e10000-2)# int e8
BigIron RX(config-if-e10000-8)# dhcp-gateway-list 3
BigIron RX(config-if-e10000-8)# int e 14
BigIron RX(config-if-e10000-14)# dhcp-gateway-list 2
```


Syntax: dhcp-gateway-list <num> <ip-addr>

Configuring Packet Parameters

You can configure the following packet parameters to control how the BigIron RX sends IP packets to other devices on an Ethernet network. The BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX.
- 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 BigIron RX 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 BigIron RX 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 MG8(config)# default-max-frame-size 2000
BigIron RX MG8(config)# write memory
BigIron RX MG8(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 18.3 shows the ports of each Interface module.

Table 18.3: Available Ports per PPCR

Module type	Number of Packet Processors (PPCR)	Ports in a PPCR			
		PPC1	PPCR2	PPCR3	PPCR4
4 x 10G	4	1	2	3	4
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 MG8(config)#interface ethernet 6/4
BigIron RX MG8(config-if-e1000-6/4)#max-frame-size 1500 bytes
BigIron RX MG8(config-if-e1000-6/4)#write memory
BigIron RX MG8(config-if-e1000-6/4)#exit
BigIron RX MG8(config)#reload
```

In this example the maximum frame size is applied to port 4 of a 40 x 1G Ethernet Interface module. That means that this maximum will apply to ports 1 to 10 on the interface module.

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 18.3. 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 BigIron RX 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 BigIron RX 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

- 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.

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.

NOTE: IP MTU can be configured globally or on physical ports; however, IP MTU cannot be applied to virtual routing interfaces. Although the **ip mtu** command may appear under the virtual routing interface level and you can enter a value for **ip mtu** at this level, the IP MTU configuration at this level is applied globally to all ports on the device. Also, the value you enter will be checked against the value of the global maximum frame size.

Changing the Router ID

In most configurations, a BigIron RX has multiple IP addresses, usually configured on different interfaces. As a result, a BigIron RX'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 BigIron RX by just one of the IP addresses configured on the BigIron RX, regardless of the interfaces that connect the BigIron RX 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 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 BigIron RX. 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 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, 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 BigIron RX 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 BigIron RX 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 BigIron RX to use a single source interface for all Telnet, TACACS/TACACS+, or RADIUS packets, the BigIron RX uses the same IP address as the source for all packets of the specified type, regardless of the port(s) 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 Foundry 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 BigIron RX.

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 BigIron RX.

```
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 BigIron RX.

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 BigIron RX.

Syntax: ip radius source-interface ethernet <slot/port> | loopback <num> | ve <num>

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

Configuring ARP Parameters

Address Resolution Protocol (ARP) is a standard IP protocol that enables the BigIron RX to obtain the MAC address of another device's interface when the BigIron RX knows the IP address of the interface. ARP is enabled by default and cannot be disabled.

How ARP Works

The BigIron RX needs to know a destination's MAC address when forwarding traffic, because the BigIron RX 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 BigIron RX. The device can be the packet's final destination or the next-hop router toward the destination.

The BigIron RX encapsulates IP packets in Layer 2 packets regardless of whether the ultimate destination is locally attached or is multiple router hops away. Since the BigIron RX's IP route table and IP forwarding cache contain IP address information but not MAC address information, the BigIron RX cannot forward IP packets based solely on the information in the route table or forwarding cache. The BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX does the following:

- First, the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX, the device sends an ARP response containing its MAC address. The response is a unicast packet addressed directly to the BigIron RX. The BigIron RXplaces 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 BigIron RX. A MAC broadcast is not routed to other networks. However, some routers, including the BigIron RX, can be configured to reply to ARP requests from one network on behalf of devices on another network. See "Enabling Proxy ARP" on page 18-25.

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 BigIron RX 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 BigIron RX 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)# rate-limit-arp 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] rate-limit-arp <num>

The <num> parameter specifies the number of ARP packets and can be from 0 – 100. If you specify 0, the device will not accept any ARP packets.

NOTE: If you want to change a previously configured the ARP rate limiting policy, you must remove the previously configured policy using the **no rate-limit-arp** <num> command before entering the new policy.

Changing the ARP Aging Period

When the BigIron RX places an entry in the ARP cache, the BigIron RX 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 BigIron RX, 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 BigIron RX 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 BigIron RX connected to two subnets, 10.10.10.0/24 and 20.20.20.0/24, the BigIron RX 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 BigIron RX 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 BigIron RX, 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 Foundry device receives an ARP request from the device that has the entry's address.

You can increase the number of configurable static ARP entries. See "Changing the Maximum Number of Entries the Static ARP Table Can Hold" on page 18-26.

To display the ARP cache and static ARP table, see the following:

- To display the ARP table, see "Displaying the ARP Cache" on page 18-50.
- To display the static ARP table, see "Displaying the Static ARP Table" on page 18-51.

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 BigIron RX.

Syntax: arp <num> <ip-addr> <mac-addr> ethernet <slot/port>

The <num> parameter specifies the entry number. It can be from 1 up to the maximum number of static entries allowed on the device.

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 BigIron RX 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. See "Displaying and Modifying System Parameter Default Settings" on page 4-13.

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 8000
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).

Configuring Forwarding Parameters

The following configurable parameters control the forwarding behavior of the BigIron RX:

- 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 BigIron RX.

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 BigIron RX 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

Foundry 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 BigIron RX supports both types of IP source routing:

- Strict source routing – requires the packet to pass through only the listed routers. If the BigIron RX receives a strict source-routed packet but cannot reach the next hop interface specified by the packet, the BigIron RX discards the packet and sends an ICMP Source-Route-Failure message to the sender.

  NOTE: The BigIron RX allows you to disable sending of the Source-Route-Failure messages. See "Disabling ICMP Messages" on page 18-28.

- 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 BigIron RX 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 BigIron RX treats IP packets with all ones in the host portion of the address as IP broadcast packets. For example, the BigIron RX 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 BigIron RX).

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 BigIron RX to treat IP packets with all zeros in the host portion of the destination IP address as broadcast packets.

NOTE: When you enable the BigIron RX for zero-based subnet broadcasts, the BigIron RX still treats IP packets with all ones the host portion as IP subnet broadcasts too. Thus, the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX replies to IP pings from other IP devices.
- Destination Unreachable messages – If the BigIron RX receives an IP packet that it cannot deliver to its destination, the BigIron RX 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 BigIron RX.

Disabling Replies to Broadcast Ping Requests

By default, the BigIron RX 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 BigIron RX 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 BigIron RX's response to the following types of ICMP Unreachable messages:

- Administration – The packet was dropped by the Foundry 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 BigIron RX cannot forward the packet without fragmenting it.
- Host – The destination network or subnet of the packet is directly connected to the BigIron RX, but the host specified in the destination IP address of the packet is not on the network.
- Network – The BigIron RX 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 BigIron RX, 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 BigIron RX from sending these types of ICMP messages on an individual basis.

NOTE: Disabling an ICMP Unreachable message type does not change the BigIron RX'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 BigIron RX 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 BigIron RX automatically creates a route for the network the interface is in.
- RIP – If RIP is enabled, the BigIron RX can learn about routes from the advertisements other RIP routers send to the BigIron RX. If the route has a lower administrative distance than any other routes from different sources to the same destination, the BigIron RX 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 BigIron RX uses if other default routes to the destination are not available. See “Configuring a Default Network Route” on page 18-36.
- 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 BigIron RX interface through which you want the BigIron RX 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 BigIron RX drops traffic forwarded to the null interface.

The following parameters are optional:

- The route’s metric – The value the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX can load balance traffic to the routes' destination. For information about IP load balancing, see "Configuring IP Load Sharing" on page 18-38.
- Path redundancy – When you add multiple static IP routes for the same destination, but give the routes different metrics or administrative distances, the BigIron RX 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 18-34
- "Configuring Standard Static IP Routes and Interface or Null Static Routes to the Same Destination" on page 18-34

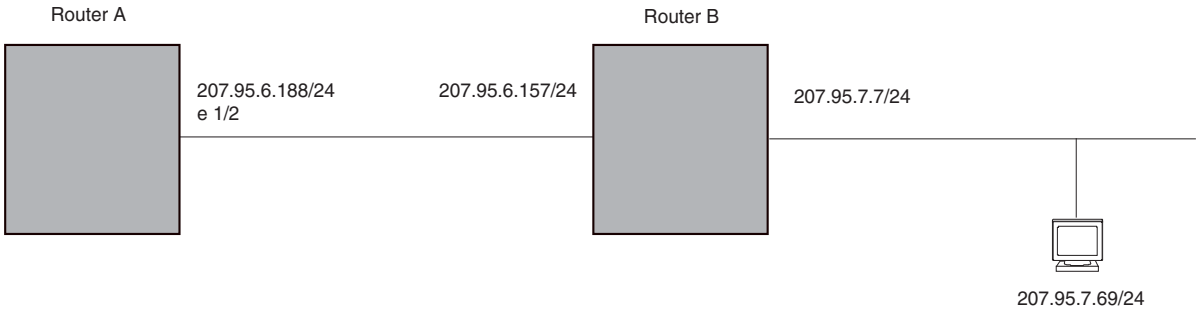
Static Route States Follow Port States

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 BigIron RX to adjust to changes in network topology. The BigIron RX does not continue trying to use routes on unavailable paths but instead uses routes only when their paths are available.

Figure 18.6 shows a network containing a static route. The static route is configured on Router A, as shown in the CLI following the figure.

Figure 18.6 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 BigIron RX interface through which the BigIron RX can reach the route. The BigIron RX 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 BigIron RX 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 BigIron RX. The <num> parameter is a virtual interface number. The <slot/port> is the port's number of the BigIron RX. If you specify an Ethernet port, the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX will replace the static route if it receives a route with a lower administrative distance. See "Changing Administrative Distances" on page 27-21 for a list of the default administrative distances for all types of routes.

Configuring a "Null" Route

You can configure the BigIron RX 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 BigIron RX receives a packet destined for the address, the BigIron RX 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 BigIron RX 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 BigIron RX's CPU.

Hardware dropping for IP traffic sent to the null0 interface is supported.

You can optionally configure the BigIron RX 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 BigIron RX 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

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 BigIron RX 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 BigIron RX alternates between the two routes. For information about IP load balancing, see "Configuring IP Load Sharing" on page 18-38.
- 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 BigIron RX will always use the route with the lowest metric. If this route becomes unavailable, the BigIron RX will fail over to the static route with the next-lowest metric, and so on.

NOTE: You also can bias the BigIron RX 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, see "Changing Administrative Distances" on page 27-21.

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 BigIron RX 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, see "Configuring a Static IP Route" on page 18-32.

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 BigIron RX has multiple routes to the same destination, the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 18.7 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 BigIron RX always prefers the static route with the lower metric. In this example, the BigIron RX 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 BigIron RX sends traffic to the null route instead.

Figure 18.7 Standard and null static routes to the same destination network

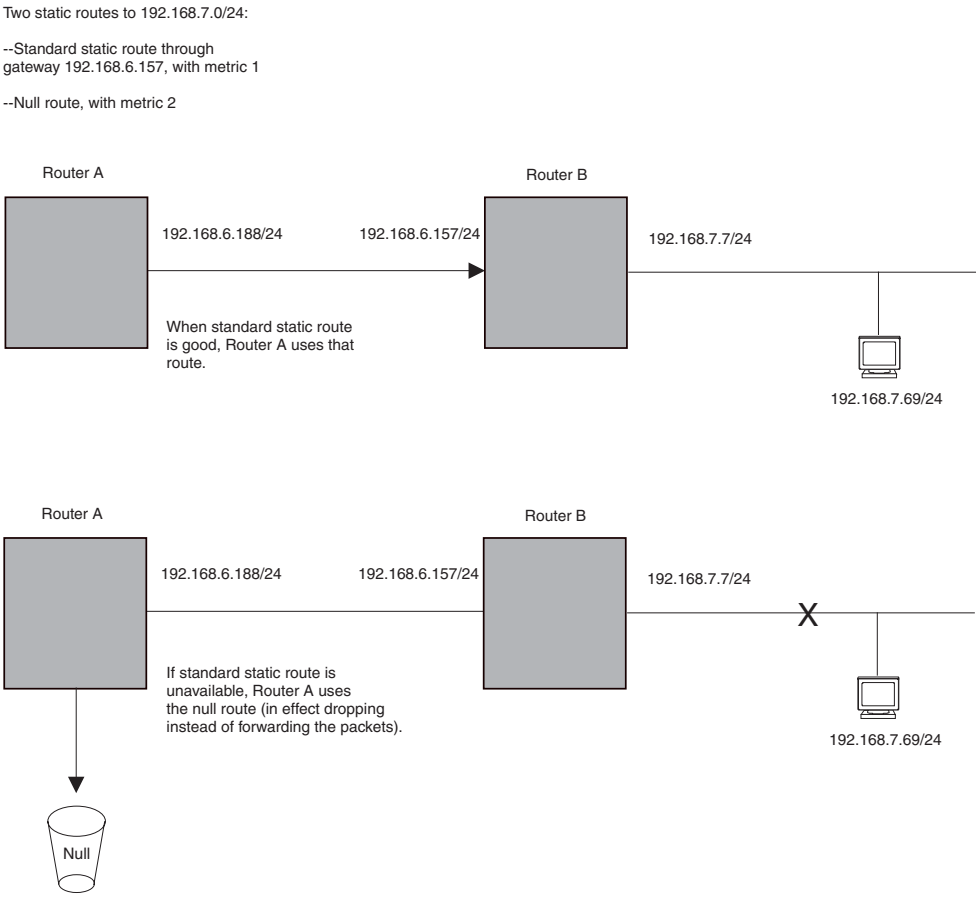
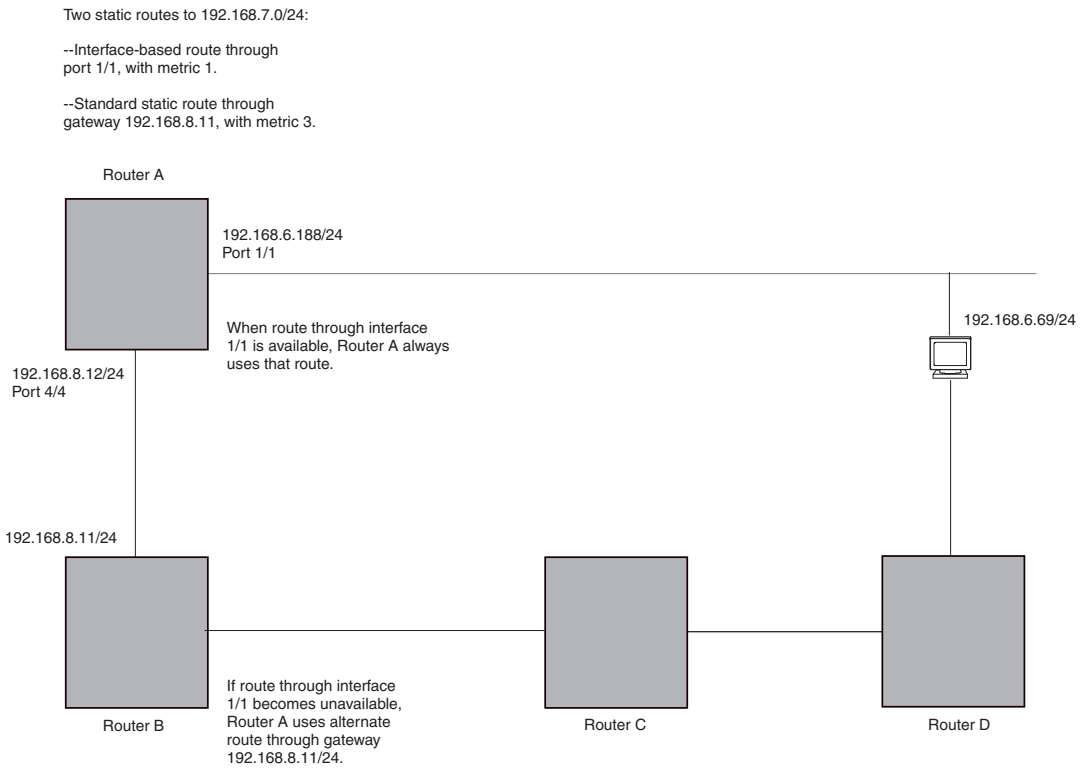


Figure 18.8 shows another example of two static routes. A standard static route and an interface-based static route are configured for destination network 192.168.6.0/24. The interface-based static route has a lower metric than the standard static route. As a result, the BigIron RX always prefers the interface-based route when the route is available. However, if the interface-based route becomes unavailable, the BigIron RX still forwards the traffic toward the destination using an alternate route through gateway 192.168.8.11/24.

Figure 18.8 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 18.7 on page 18-35, 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 BigIron RX 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, see “Configuring a Static IP Route” on page 18-32.

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 BigIron RX to always prefer this route when it is available. If the route becomes unavailable, the BigIron RX uses an alternate route through the next-hop gateway 192.168.8.11/24.

Configuring a Default Network Route

The BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX uses ***IP load sharing*** to select a path to the destination.[1]

IP load sharing is based on the destination address of the traffic. BigIron RX supports load sharing based on individual host addresses or on network addresses.

You can enable a BigIron RX to load balance across up to eight equal-cost paths. The default maximum number of equal-cost load sharing paths is four.

NOTE: IP load sharing is not based on source routing, only on next-hop routing.

NOTE: The term “path” refers to the next-hop router to a destination, not to the entire route to a destination. Thus, when the software compares multiple equal-cost paths, the software is comparing paths that use different next-hop routers, with equal costs, to the same destination.

In many contexts, the terms “route” and ”path” mean the same thing. Most of the user documentation uses the term “route” throughout. The term “path” is used in this section to refer to an individual next-hop router to a destination, while the term “route” refers collectively to the multiple paths to the destination. Load sharing applies when the IP route table contains multiple, equal-cost paths to a destination.

NOTE: The BigIron RX also performs load sharing among the ports in aggregate links. See “Trunk Group Load Sharing” on page 6-4, in the “Configuring Trunk Groups” chapter.

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 BigIron RX 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 BigIron RX 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 BigIron RX:

- 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

1.IP load sharing is also called “Equal-Cost Multi-Path (ECMP)” load sharing or just “ECMP”

- 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. See 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 BigIron RX 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 BigIron RX 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. See "Configuring Load Balancing and Redundancy Using Multiple Static Routes to the Same Destination" on page 18-34.
- 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 18.4 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 BigIron RX, load sharing for static IP routes, RIP routes, OSPF routes, and BGP4 routes also is enabled by default.

Table 18.4: Default Load Sharing Parameters for Route Sources

Route Source	Default Maximum Number of Paths	Maximum Number of Paths	See...
Static IP route	4[a]	8[a]	18-40
RIP	4[a]	8[a]	18-40
OSPF	4	8	18-40
BGP4	1	4	27-41

a.This value depends on the value for IP load sharing, and is not separately configurable.

How IP Load Sharing Works

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.

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 BigIron RX 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 BigIron RX 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. (See “How IP Load Sharing Works” on page 18-40.)

Configuring IRDP

The BigIron RX uses ICMP Router Discovery Protocol (IRDP) to advertise the IP addresses of its router interfaces to directly attached hosts. IRDP is disabled by default. You can enable it globally or on individual ports.

- 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 BigIron RX periodically sends Router Advertisement messages out the IP interfaces on which the feature is enabled. The messages advertise the BigIron RX's IP addresses to directly attached hosts who listen for the messages. In addition, hosts can be configured to query the BigIron RX 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 BigIron RX responds to the Router Solicitation messages. Some clients interpret this response to mean that the BigIron RX is the default gateway. If another router is actually the default gateway for these clients, leave IRDP disabled on the BigIron RX.

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.

- Packet type – The BigIron RX 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 BigIron RX sends the Router Advertisement messages every 450 – 600 seconds by default. The time within this interval that the BigIron RX 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 BigIron RX 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 BigIron RX uses to send Router Advertisement.

- **broadcast** – The BigIron RX sends Router Advertisement as IP broadcasts. This is the default.
- **multicast** – The BigIron RX 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 BigIron RX should consider the advertisement to be valid. When a host receives a new Router Advertisement message from the BigIron RX, the host resets the hold time for the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX. 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 BigIron RX 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 BigIron RX forwards client requests for any of the application ports the BigIron RX 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 BigIron RX 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 BigIron RX to forward BootP/DHCP requests, see "Configuring BootP/DHCP Forwarding Parameters" on page 18-44.

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 BigIron RX is not also disabled.

Enabling Forwarding for a UDP Application

If you want the BigIron RX to forward client requests for UDP applications that the BigIron RX 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 BigIron RX cannot forward the requests unless you configure the helper address. See "Configuring an IP Helper Address" on page 18-45.

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
```

This command disables forwarding of SNMP requests to the helper addresses configured on BigIron RX 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 BigIron RX is enabled to forward, the BigIron RX 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

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 BigIron RX 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 BigIron RX does not forward the request.

You can configure the BigIron RX 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.

BootP/DHCP Forwarding Parameters

The following parameters control the BigIron RX'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 BigIron RX cannot forward a request to the server unless you configure a helper address for the server.
- Gateway address – The BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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. See “Configuring an IP Helper Address” on page 18-44.

Changing the IP Address Used for Stamping BootP/DHCP Requests

When the BigIron RX forwards a BootP/DHCP request, the BigIron RX “stamps” the Gateway Address field. The default value the BigIron RX 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 BigIron RX will place this IP address in the Gateway Address field of BootP/DHCP requests that the BigIron RX 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 BigIron RX receives a BootP/DHCP request, the BigIron RX 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 BigIron RX allows, the BigIron RX increments the hop count by one and forwards the request.
- If the hop count is greater than the maximum hop count the BigIron RX allows, the BigIron RX 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 BigIron RX to forward BootP/DHCP requests that have passed through up to ten previous hops before reaching the BigIron RX.

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 – see “Displaying Global IP Configuration Information” on page 18-46.
- IP interfaces – see “Displaying IP Interface Information” on page 18-48.
- ARP entries – see “Displaying ARP Entries” on page 18-50.
- Static ARP entries – see “Displaying ARP Entries” on page 18-50.
- IP forwarding cache – see “Displaying the Forwarding Cache” on page 18-52.
- IP route table – see “Displaying the IP Route Table” on page 18-54.
- IP traffic statistics – see “Displaying IP Traffic Statistics” on page 18-57.

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 – see “Displaying RIP Filters” on page 25-9.
- OSPF information – see “Displaying OSPF Information” on page 26-32.
- BGP4 information – see “Displaying BGP4 Information” on page 27-69.
- DVMRP information – see “Displaying Information About an Upstream Neighbor Device” on page 24-35
- PIM information – see “Displaying PIM Sparse Configuration Information and Statistics” on page 24-17.
- VRRP or VRRPE information – see “Displaying VRRP and VRRPE Information” on page 15-16.

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 18.5: 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 BigIron RX. If the packet's TTL value is higher than the value specified in this field, the Foundry router drops the packet. To change the maximum TTL, see "Changing the TTL Threshold" on page 18-26.
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, see "Changing the ARP Aging Period" on page 18-24.
bootp-relay-max-hops	The maximum number of hops away a BootP server can be located from the Foundry router and still be used by the router's clients for network booting. To change this value, see "Changing the Maximum Number of Hops to a BootP Relay Server" on page 18-45.
router-id	The 32-bit number that uniquely identifies the Foundry router. By default, the router ID is the numerically lowest IP interface configured on the router. To change the router ID, see "Changing the Router ID" on page 18-21.
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 Foundry 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 Foundry routers is 1. To list the default administrative distances for all types of routes or to change the administrative distance of a static route, see "Changing Administrative Distances" on page 27-21.
Policies	

Table 18.5: CLI Display of Global IP Configuration Information

This Field...	Displays...
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.
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 18.6: CLI Display of Interface IP Configuration Information

This Field...	Displays...
Interface	The type and the slot and port number of the interface.
IP-Address	The IP address of the interface. **Note**: If an "s" is listed following the address, this is a secondary address. When the address was configured, the interface already had an IP address in the same subnet, so the software required the "secondary" option before the software could add the interface.
OK?	Whether the IP address has been configured on the interface.
Method	Whether the IP address has been saved in NVRAM. If you have set the IP address for the interface in the CLI, but have not saved the configuration, the entry for the interface in the Method field is "manual".
Status	The link status of the interface. If you have disabled the interface with the **disable** command, the entry in the Status field will be "administratively down". Otherwise, the entry in the Status field will be either "up" or "down".
Protocol	Whether the interface can provide two-way communication. If the IP address is configured, and the link status of the interface is up, the entry in the protocol field will be "up". Otherwise the entry in the protocol field will be "down".

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 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
```

Displaying ARP Entries

You can display the ARP cache and the static ARP table. The ARP cache contains entries for devices attached to the BigIron RX. 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 18.7: 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 BigIron RX learned the entry from an incoming packet. • Static – The BigIron RX loaded the entry from the static ARP table when the device for the entry was connected to the BigIron RX.
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, see "Displaying Global IP Configuration Information" on page 18-46. To change the ARP aging interval, see "Changing the ARP Aging Period" on page 18-24. **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.

NOTE: 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, see the syntax of the **show arp** command under “Displaying the ARP Cache” on page 18-50.

Table 18.8: 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, see “Changing the Maximum Number of Entries the Static ARP Table Can Hold” on page 18-26.
Index	The number of this entry in the table. You specify the entry number when you create the entry.
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 18.9: 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 Foundry 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 Foundry 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 VLAN(s) 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
```

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 BigIron RX.

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 BigIron RX:

```
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 18.10: CLI Display of IP Route Table

This Field...	Displays...
Destination	The destination network of the route.
NetMask	The network mask of the destination address.

Table 18.10: CLI Display of IP Route Table (Continued)

This Field...	Displays...
Gateway	The next-hop router.
Port	The port through which this router sends packets to reach the route's destination.
Cost	The route's cost.

Type	The route type, which can be one of the following:
	• B – The route was learned from BGP.
	• D – The destination is directly connected to this BigIron RX.
	• 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.

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 BigIron RX, 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> show ip traffic

IP Statistics

  139 received, 145 sent, 0 forwarded
  0 filtered, 0 fragmented, 0 reassembled, 0 bad header
  0 no route, 0 unknown proto, 0 no buffer, 0 other errors

ICMP Statistics
Received:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp reply, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation
Sent:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp reply, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation

UDP Statistics
  1 received, 0 sent, 1 no port, 0 input errors

TCP Statistics
  0 active opens, 0 passive opens, 0 failed attempts
  0 active resets, 0 passive resets, 0 input errors
  138 in segments, 141 out segments, 4 retransmission

RIP Statistics
  0 requests sent, 0 requests received
  0 responses sent, 0 responses received
  0 unrecognized, 0 bad version, 0 bad addr family, 0 bad req format
  0 bad metrics, 0 bad resp format, 0 resp not from rip port
  0 resp from loopback, 0 packets rejected
```

Syntax: show ip traffic

The **show ip traffic** command displays the following information.

Table 18.11: 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.

Table 18.11: CLI Display of IP Traffic Statistics

This Field...	Displays...
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 Foundry customer support.
other errors	The number of packets that this device dropped due to error types other than the types listed above.
ICMP statistics The ICMP statistics are derived from RFC 792, “Internet Control Message Protocol”, RFC 950, “Internet Standard Subnetting Procedure”, and RFC 1256, “ICMP Router Discovery Messages”. Statistics are organized into Sent and Received. The field descriptions below apply to each.	
total	The total number of ICMP messages sent or received by the device.
errors	This information is used by Foundry 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.

Table 18.11: CLI Display of IP Traffic Statistics

This Field...	Displays...
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 Foundry 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 Foundry 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.
input errors	This information is used by Foundry 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.

Table 18.11: CLI Display of IP Traffic Statistics

This Field...	Displays...
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 Foundry 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 Foundry 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 Foundry customer support.
resp from loopback	The number of RIP responses received from loopback interfaces.
packets rejected	This information is used by Foundry customer support.

Chapter 19
Configuring Rate Limiting

You can configure an inbound port on the BigIron RX to use one of the following rate limiting policies:

- Port-based – Limits the rate of inbound traffic on a physical port to a specified rate. See “Configuring a Port-Based Rate Limiting Policy” on page 19-2.
- Port-and-priority-based – Limits the rate on a hardware forwarding queue on a physical port. See “Configuring a Port-and-Priority-Based Rate Limiting Policy” on page 19-3.
- Port-and-VLAN-based – Limits the rate of packets tagged with a specific VLAN on a physical port. Only one rate can be specified for each VLAN. Up to 10 VLAN-based policies can be configured for a port. See “Configuring a Port-and-VLAN-Based Rate Limiting Policy” on page 19-3.)
- 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. Up to 1024 VLAN Group-based policies can be configured for a port under normal conditions or 4096 policies if priority-based rate limiting is disabled. See “Configuring a VLAN-Group-Based Rate Limiting Policy” on page 19-3.
- Port-and-ACL-based – Limits the rate of inbound traffic on a physical port that matches the permit conditions in Access Control Lists (ACLs). See “Configuring a Port-and-ACL-Based Rate Limiting Policy” on page 19-5.
- Port-and-IPV6-ACL-based – Limits the rate of traffic on an individual physical port that matches the permit conditions of IPV6 ACL. See “Configuring a Port-and-IPv6 ACL-Based Rate Limiting Policy” on page 19-5.

Rate Limiting Parameters and Algorithm

A rate limiting policy specifies two parameters: average rate and maximum burst.

Average Rate

The *Average Rate* is the maximum number of bits a port is allowed to receive during a one-second interval. The rate of the traffic that matches the rate limiting policy will not exceed the average rate.

The Average Rate represents a percentage of an interface's line rate (bandwidth), expressed in bits per second (bps). It cannot be smaller than 515,624 bits per second (bps) and it cannot be larger than the port’s line rate.

Average 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. For example, if you enter 600,000 bps, the value will be adjusted to 515,624 bps. The adjusted rate is sometimes called the *adjusted average rate*.

Maximum Burst

Maximum burst provides a higher than average rate to traffic that meet the rate limiting criteria. The maximum burst rate cannot be smaller than 65536 bits.

Configuration Considerations

- Rate limiting policies apply only to inbound ports on the BigIron RX.
- 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.
- When a port-and-VLAN-based rate limiting policy is applied to a port, all the ports controlled by the same packet processor are rate limited for that VLAN. You cannot apply a port-and-VLAN-based rate limiting policy on another port of the same packet processor for the same VLAN ID.
- 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 average rate in a rate limiting policy cannot be less than 515,624 bits per second, must be in multiples of 515,624, and cannot be more than the port's line rate.
- The maximum burst in a rate limit policy can be less than the average rate, but cannot be less than 65536 bits and cannot be more than the port's line rate.
- Control packets are not subject to rate limiting.
- You cannot create a trunk if any of the physical ports that are members of the trunk has a rate limiting policy.
- 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 average 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 average rate to be less than the expected average 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

Configuring Rate Limiting Policies on the BigIron RX

Configuring rate-limiting policies involves using the rate-limit command and specifying the average rate and maximum burst at the interface level of the CLI.

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 in 500000000 750000000
Average rate is adjusted to 499639656 bits per second
```

The commands configure a rate limiting policy for inbound traffic on port 1/1. The policy limits the average rate of all inbound traffic to 499639656 bps with a maximum burst size of 750 Mbps.

Syntax: [no] rate-limit input <average-rate> <maximum-burst>

input applies rate limiting to inbound traffic on the port. **Input** can be abbreviated as **in**.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software adjusts the number you enter to the lower multiple of 515,624 bps. See "Average Rate" on page 19-1 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. See "Maximum Burst" on page 19-1 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 Foundry 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 limit the rate on hardware forwarding queues 0 and 1 on the port to an adjusted average rate of 499321856 bps with a maximum burst size of 750 Mbits.

Syntax: [no] rate-limit input priority <num> <average-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, see “Assigning QoS Priorities to Traffic” on page 16-5.

For information on the other parameters, see “Configuring a Port-Based Rate Limiting Policy” on page 19-2.

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 average rate of inbound traffic on port 1/3. The first policy limits packets with VLAN tag 10 to an average rate of 499321856 bps with a maximum burst size of 750 Mbits on the port. The second policy limits packets with VLAN tag 20 to an average rate of 97523712 bps 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> <average-rate> <maximum-burst>

The vlan <vlan-number> parameter species the VLAN ID to which the policy applies. Refer to “Configuration Considerations” on page 19-2 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 19-2.

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 average rate of inbound traffic (packets tagged with VLANs 3, 5, 6, or 7 from VLAN group 10) from VLAN group 10 to an adjusted average rate of 499321856 bps with a maximum burst size of 750 Mbits.

Syntax: rate-limit in group <group-number> <average-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 19-2.

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. Rates for queues 2 and 3 are limited to an adjusted average rate of 499321856 bps with a maximum burst size of 750 Mbits.

Syntax: rate-limit in group <group-number> priority <num> <average-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, see “Assigning QoS Priorities to Traffic” on page 16-5.

For information on the average rate and maximum burst, refer to “Configuring a Port-Based Rate Limiting Policy” on page 19-2.

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 Rate Limiting 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.

See the section “Access Control List” on page 21-1 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 **deny** condition are not subject to rate limiting and allowed to pass through. Refer to “Dropping Traffic Denied by a Rate Limiting ACL” on page 19-5 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 average rate of all inbound IP traffic that match the permit rules of ACLs 50 and 60. The first policy limits the rate of all permitted IP traffic from host 1.1.1.2 to an average rate of `499321856` bps with a maximum burst size of 750 bits. Rate of all traffic from host 1.1.1.3 is not subject to rate limiting since it is denied by ACL 50; it is merely forwarded on the port.

The second policy limits the rate of all IP traffic from host 2.2.2.3 to an average rate of `97523712` bps 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> <average-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 19-2.

For information on the number of ACL-based rate limiting policies that can be configured, refer to the “Configuration Considerations” on page 19-2.

Dropping Traffic Denied by a Rate Limiting ACL

With the strict ACL feature, you can configure a port to drop the traffic that matched an ACL deny filter in a port-and-ACL-based rate limiting policy. For example, enter the following command at the interface level:

```
BigIron RX(config-if-e1000-1/5)# rate-limit strict-acl
```

Syntax: [no] rate-limit strict-acl

Configuring a Port-and-IPv6 ACL-Based Rate Limiting Policy

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 average rate of all inbound IP traffic that matches the permit rules of `sample` to an average 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> <average-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, see “Configuring a Port-Based Rate Limiting Policy” on page 19-2.

Displaying Rate Limiting Policies

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 20
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 Foundry 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.

This chapter presents the following information:

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 20-2

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.
- BigIron RX processes ACLs in hardware.
- By default, the device processes broadcast traffic in software. Filtering of broadcast packets is not handled by the hardware.
- You can use Layer 2 ACLs to block management access to the BigIron RX. For example, you can use a Layer 2 ACL clause to block a certain host from establishing a connection to the device through Telnet.

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 BigIron RX evaluates traffic coming into the port against each ACL clause. When a match occurs, the BigIron RX 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 appletalk
BigIron RX(config)# access-list 400 deny any any any etype ipx-raw
BigIron RX(config)# access-list 400 deny any any any etype ipx-snap
BigIron RX(config)# access-list 400 deny any any any etype ipx-1lc
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 AppleTalk and IPX traffic, and permit all other traffic in VLAN 100.

Here is another example:

```
BigIron RX(config)# access-list 400 deny any etype arp
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 traffic and permit all other traffic in VLAN 100.

For more examples of valid Layer 2 ACL clauses, see “Example Layer 2 ACL Clauses” on page 20-3.

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 don't need to specify a mask and the clause matches on all MAC addresses.

The <dest-mac> <mask> | **any** parameter specifies the destination MAC address. The syntax rules are the same as those for the <src-mac> <mask> | **any** parameter.

The optional <vlan-id> | **any** parameter specifies the vlan-id to be matched against the vlan-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-l5 (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, for the BigIron MG8 and NetIron 40G, 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 for the BigIron and NetIron 40G:

```
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. Enter a command such as the following at the Interface level of the CLI:

```
NetIron4000(config)# int e 4/12
NetIron4000(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:

```
NetIron4000(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

This chapter discusses 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, see “Layer 2 ACLs” on page 20-1.

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, not globally. Otherwise, a loop can occur. See the chapters for a specific feature for information on using ACLs as input to those features.

This chapter presents the following sections:

How the BigIron RX Processes ACLs

The BigIron RX processes traffic that ACLs filter in hardware. The BigIron RX creates and entry for each ACL in the Content Addressable Memory (CAM) at startup or when the ACL is created. The BigIron RX 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 ACLs, but you can have up to 8,000 statements (rules) in all the ACL configurations on the device.
- A port supports only one ACL; however, the ACL can contain multiple statements. For example, both ACLs 101 and 102 on cannot be supported 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. See “Applying an ACLs to Interfaces” on page 21-22.
- 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)

Disabling or Re-Enabling Access Control Lists (ACLs)

The ACL feature is always enabled on BigIron RX; it cannot be disabled.

Default ACL Action

The default action when no ACLs is configured on a BigIron RX is to permit all traffic. However, once you configure an ACL and apply it to a port, the default action for that port is to deny all traffic that is not explicitly permitted on the port.

- If you want to tightly control access, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.
- If you want to secure access in environments with many users, you might want to configure ACLs that consist of explicit deny entries, then add an entry to permit all access to the end of each ACL. The software permits packets that are not denied by the deny entries.

NOTE: Do not apply an empty ACL (an ACL ID without any corresponding entries) to an interface. If you accidentally do this, the software applies the default ACL action, deny all, to the interface and thus denies all traffic.

Types of IP ACLs

IP ACLs can be configured as standard or extended ACLs. A standard ACL permits or denies packets based on source IP address. An extended ACL permits or denies packets based on source and destination IP address and also based on IP protocol information.

Standard or extended ACLs can be numbered or named. Standard numbered ACLs have an idea of 1 – 99. Extended numbered ACLs are numbered 100 – 199. IDs for standard or extended ACLs can be a character string. In this document, ACLs 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 (for a standard ACL) or 100 – 199 (for an extended ACL) or a character string. The ACL ID identifies a collection of individual ACL entries. When you apply ACL entries to an interface, you do so by applying the ACL ID that contains the ACL entries to the interface, instead of applying the individual entries to the interface. This makes applying large groups of access filters (ACL entries) to interfaces simple.

  NOTE: This is different from IP access policies. If you use IP access policies, you apply the individual policies to interfaces.

- ACL entry – An ACL entry are the filter commands associated with an ACL ID. These are also called "statements". The maximum number of ACL entries you can configure is a system-wide parameter and depends on the BigIron RX you are configuring. You can configure up to the maximum number of entries in any combination in different ACLs. The total number of entries in all ACLs cannot exceed the system maximum.

You configure ACLs on a global basis, then apply them to the incoming or outgoing traffic on specific ports. You can apply only one ACL to a port's inbound traffic and only one ACL to a port's outbound traffic. The software applies the entries within an ACL in the order they appear in the ACL's configuration. As soon as a match is found, the software takes the action specified in the ACL entry (permit or deny the packet) and stops further comparison for that packet.

Enabling Support for Additional ACL Statements

You can enable support for additional ACL statements if the BigIron RX has enough space to hold 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 flash card or TFTP server, then loading them using one of the following commands:

- **copy slot1 | slot2 running** <from-name>
- **ncopy slot1 | slot2** <from-name> **running**
- **copy tftp running-config** <ip-addr> <filename>
- **ncopy tftp** <ip-addr> <from-name> **running-config**

In this case, the ACLs are added to the existing configuration.

Configuring Numbered and Named ACLs

When you configure ACLs, you can refer to the ACL by a numeric ID or by an alphanumeric name. The commands to configure numbered ACLs are different from the commands for named ACLs.

- If you refer to the ACL by a numeric ID, you can use 1 – 99 for a standard ACL or 100 – 199 for an extended ACL. This document refers to this ACL as *numbered ACL*.
- If you refer to the ACL by a name, you specify whether the ACL is a standard ACL or an extended ACL, then specify the name. This document refers to this ACL type as *named ACL*.

You can configure up to 100 standard numbered IP ACLs and 100 extended numbered IP ACLs. You also can configure up to 100 standard named ACLs and 100 extended named ACLs by number. Regardless of how many ACLs you have, the BigIron RX can have 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, see “Configuring Standard or Extended Named ACLs” on page 21-15.
- For configuration information on extended ACLs, see “Configuring Extended Numbered ACLs” on page 21-5.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard ACLs. There is no limit to the number of ACL entries an ACL can contain except for the system-wide limitation. For the number of ACL entries supported on a BigIron RX, see “ACL IDs and Entries” on page 21-3.

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

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 BigIron RX to generate Syslog entries and SNMP traps for packets that are permitted or denied by the access policy. If you use the **log** argument, the ACL entry is sent to the CPU for processing. See the section "ACL Logging" on page 21-23 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, see "Configuring Numbered and Named ACLs" on page 21-3.
- For configuration information on standard ACLs, see "Configuring Standard Numbered ACLs" on page 21-4.

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 BigIron RX 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 BigIron RX 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. U

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

General Parameters for Extended ACLs

The following parameters apply to any extended ACL you are creating.

<num>	Enter 100 – 199 for an extended ACL.
deny | **permit**	Enter **deny** if the packets that match the policy are to be dropped; **permit** if they are to be forwarded.
<ip-protocol>	Indicate the type of IP packet you are filtering. You can specify a well-known name for any protocol whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI.
<source-ip> | <hostname>	Specify the source IP host for the policy. If you want the policy to match on all source addresses, enter **any**.
<wildcard>	Specifies the portion of the source IP host address to match against. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy. If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file. The IP subnet masks in CIDR format is saved in the file in "/<mask-bits>" format. If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show access-list** command.
<destination-ip> | <hostname>	Specify the destination IP host for the policy. If you want the policy to match on all destination addresses, enter **any**.
fragment	Enter this keyword if you want to filter fragmented packets. See "Enabling ACL Filtering of Fragmented or Non-Fragmented Packets" on page 21-28. **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. See “Enabling ACL Filtering of Fragmented or Non-Fragmented Packets” on page 21-28.

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. See “Enabling ACL Filtering of Fragmented or Non-Fragmented Packets” on page 21-28.

fragment-offset <number> — Enter this parameter if you want to filter a specific fragmented packets. Enter a value from 0 – 8191. See “Enabling ACL Filtering of Fragmented or Non-Fragmented Packets” on page 21-28.

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. See “ACL Logging” on page 21-23.

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. See 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 BigIron RX 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. See “Configuring Numbered and Named ACLs” on page 21-3.

precedence <name> | <num>

The precedence option for of an IP packet is set in a three-bit field following the four-bit header-length field of the packet's header. 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>

This parameter filters packets based on their IPSEC Security Parameters Index (SPI). Enter this value in hexadecimal. the range is 00000000 – ffffffff.

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.

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:

  ```
  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 21.1, the new QoS value for the packet is:

  – 802.1p (COS) value: 7
  – DSCP value: 48
  – Internal Forwarding Priority: 0

Table 21.1: 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 21.1, 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, see "Configuring Quality of Service" on page 16-1.

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-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL 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)# permit any
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, see "Configuring Standard Numbered ACLs" on page 21-4.

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 extended | 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 <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 21-4.

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]

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 21-5.

Displaying ACL Definitions

To display the ACLs configured on a BigIron RX, 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's number for the <number> parameter:

- 1 – 99 for standard ACLs
- 100 – 199 for extended ACLs

Enter **all** if you want to display all the ACLs configured on the device.

Named ACL

For a named ACL, enter a command such as the following:

```
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>

The ACL's name for the <acl-name> parameter or the ACL's 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 BigIron RX 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 BigIron RX displays TCP/UDP application information in named notation.

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 ACL entry in this ACL and is therefore denied, the software does not compare the packet to the remaining ACL entries. In this example, packets from host 209.157.22.26 will always be dropped, even though packets from this host match the second entry.

You can use the CLI to reorder entries within an ACL by individually removing the ACL entries and then re-adding them. To use this method, enter "**no**" followed by the command for an ACL entry, and repeat this for each ACL entry in the ACL you want to edit. After removing all the ACL entries from the ACL, re-add them.

This method works well for small ACLs such as the example above, but can be impractical for ACLs containing many entries. Therefore, the BigIron RX provides an alternative method. The alternative method lets you upload an ACL list from a TFTP server and replace the ACLs in the BigIron RX's running-config file with the uploaded list. Thus, to change an ACL, you can edit the ACL on the file server, then upload the edited ACL to the BigIron RX. You then can save the changed ACL to the BigIron RX'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 BigIron RX 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 BigIron RX 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 BigIron RX, 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 BigIron RX, 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 BigIron RX'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 BigIron RX reads only the ACL information in the file and ignores other commands, including **ip access-group** commands. To assign ACLs to interfaces, use the CLI.

Adding or Deleting a Comment

You can add or delete comments to an 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
permit any
```

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. Entering a **show access-list** command displays the following:

```
BigIron RX(config)# show access-list 99
Standard IP access-list 99
deny host 1.2.4.5
permit host 5.6.7.8
permit any

ACL Remarks: Permit all users
```

Syntax: [no] access-list <acl-num> remark <comment-text>

Simply entering **access-list** <acl-num> **remark** <comment-text> adds a remark to the next ACL entry you create.

The **remark** <comment-text> adds a comment to the ACL entry. The remark can have up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same command. Also, in order for the remark to be displayed correctly in the output of **show** commands, the comment must be entered immediately before the ACL entry it describes.

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 21-15.

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 you 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 displays 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 standard | extended <acl-name>

Syntax: [no] remark <string>

Syntax: deny <options> | permit <options>

The **standard | extended** 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.

The **remark** <string> adds a comment to the ACL entry that you are about to create. The comment can have up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same command. Also, in order for the remark to be displayed correctly in the output of show commands, the comment must be entered immediately before the ACL entry it describes.

Enter **deny** to deny the specified traffic or **permit** to allow the specified traffic. Complete the configuration by specifying <options> for the standard or extended ACL entry. Options you can use to configure standard or extended named ACLs are discussed in the section “Configuring Standard or Extended Named ACLs” on page 21-15.

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 needed 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 21-16 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 <line-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 or 100 – 199 for extended ACLs.

You must enter the complete deny or permit statement for the <entire-deny-or-permit-statement> variable.

Complete the configuration by specifying options for the ACL entry. Options you can use to configure standard or extended numbered ACLs are discussed in “Configuring Standard Numbered ACLs” on page 21-4 and “Configuring Extended Numbered ACLs” on page 21-5.

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.

You must enter the complete deny or permit statement for the <entire-deny-or-permit-statement> variable.

Applying an ACLs to Interfaces

Configuration examples in the section “Configuring Numbered and Named ACLs” on page 21-3 show that you apply ACLs to interfaces using the **ip access-group** command. This section present additional information about applying ACLs to interfaces.

Reapplying Modified ACLs

If you make an ACL configuration change, you must reapply the ACLs to their interfaces to place the change into effect.

An ACL configuration change includes any of the following:

- Adding, changing, or removing an ACL or an entry in an ACL
- Changing a PBR policy
- Changing ToS-based QoS mappings

To reapply ACLs following an ACL configuration change, enter the following command at the global CONFIG level of the CLI:

```
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.

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>

ACL Logging

You may want the software to log entries for ACLs in the syslog. ACL logging is disabled by default; it must be explicitly enabled on a port.

When you enable logging for ACL entries, statistics for packets that match the permit or deny conditions of the ACL entries are logged. For example, if you configure a standard ACL entry to deny all packets from source address 209.157.22.26, statistics for packets that are explicitly denied by the ACL entry are logged in the Syslog buffer and in SNMP traps sent by the BigIron RX.

The first time an ACL entry permits or denies a packet, the software immediately generates a Syslog entry and SNMP trap. The software also starts a five-minute timer. The timer keeps track of all packets explicitly denied by the ACL entries. After five minutes, 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 during the previous five minutes.

If no ACL entries explicitly permit or deny packets during an entire five-minute timer interval, the timer stops. The timer restarts when an ACL entry explicitly permits or denies a packet.

NOTE: The timer for logging packets denied by Layer 2 filters is separate.

Enabling ACL Logging

To enable ACL logging on an interface, enter the following command.

```
BigIron RX(config)# interface e 1/3
BigIron RX(config-if-e10000-1/3)# enable-deny-logging
```

Syntax: [no] enable-deny-logging

Creating ACL Entries with the Log Option

Once ACL logging is enabled on an interface, you can create ACL entries that includes the **log** option if you want statistics for packets that match the statement to be logged. Enter commands such as the following:

```
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
```

Syntax: See the appropriate sections above for configuring ACLs.

Depending on how many entries have the log option and how often packets match those entries, ACL performance can be affected. Use the **log** option only when needed.

Configuring the Layer 4 Session Log Timer

You can configure the Layer 4 session log timer, which is used for keeping track of packets explicitly denied by an ACL.

When you enable logging for an ACL entry, statistics for packets that match the permit or deny conditions of the ACL entry are logged in the Syslog buffer and in SNMP traps sent by the BigIron RX. The first time an ACL entry permits or denies a packet, the software immediately generates a Syslog entry and SNMP trap. The software also starts the Layer 4 session log timer. The timer keeps track of all packets explicitly denied by the ACL entries. 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 permit or deny packets during an entire timer interval, the timer stops. The timer restarts when an ACL entry explicitly permits or denies a packet.

To store information about denied packets during the timer interval, the BigIron RX makes entries in its Layer 4 session table. If a large number of packets are denied by the ACL during the timer interval, it can consume a large portion of the BigIron RX's Layer 4 resources. To prevent this from happening, You can configure the timer interval to be a shorter length of time.

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 permits or denies a packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for packets permitted or denied by ACLs are at the warning level of the Syslog.

When the first Syslog entry for a packet permitted or denied by an ACL is generated, the software starts an ACL timer. After this, the software sends Syslog messages every one to ten 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. For more information about the timer, see “Configuring the Layer 4 Session Log Timer” on page 21-23.

NOTE: For an ACL entry to be eligible to generate a Syslog entry for permitted or 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
.
.
.
```

In this example, the two-line message at the bottom is the first entry, which the software immediately generates the first time an ACL entry permits or denies a packet. In this case, an entry in ACL 101 denied a packet. The packet was a TCP packet from host 209.157.22.198 and was destined for TCP port 80 (HTTP) on host 198.99.4.69.

When the software places the first entry in the log, the software also starts the five-minute timer for subsequent log entries. Thus, five minutes after the first log entry, the software generates another log entry and SNMP trap for denied packets.

In this example, the software generates the second log entry five minutes later.

The time stamp for the third entry is much later than the time stamps for the first two entries. In this case, no ACLs denied packets for a very long time. In fact, since no ACLs denied packets during the five-minute interval following the second entry, the software stopped the ACL log timer. The software generated the third entry as soon as the ACL denied a packet. The software restarted the five-minute ACL log timer at the same time. As long as at least one ACL entry permits or denies a packet, the timer continues to generate new log entries and SNMP traps every five minutes.

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 16-5.)
- Enabling the IP ToS-based QoS feature described in “Configuring ToS-Based QoS” on page 16-6.

NOTE: If you use an ACL on an interface, ToS-based QoS assumes that the ACLs 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 see “Using ACL QoS Options to Filter Packets” on page 21-13

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 MG8(config)# acl-duplication-check
```

Syntax: [no] acl-duplication-check

ACL Accounting

The BigIron RX monitors the number of times an ACL is used to filter incoming or outgoing traffic on an interface.This features is enabled by default and cannot be disabled.

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:

```
BigIron RX(config)#show access-list accounting brief
Collecting ACL accounting summary for VE 1 ...  Completed successfully.

ACL Accounting Summary: (ac = accumulated since accounting started)
    Int    In ACL              Total In Hit   Out ACL             Total Out Hit
    VE 1    111                473963(1s)
                               25540391(1m)
                               87014178(5m)
                               112554569(ac)
```

The display shows the following information:

This Field...	Displays...
The IP multicast traffic snooping state	The first line of the display indicates whether IP multicast traffic snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Collecting ACL accounting summary for <interface>	Shows for which interfaces the ACL accounting information was collected and whether or not the collection was successful.
Int	The ID of the interface for which the statistics are being reported.
In ACL	The ID of the ACL used to filter the incoming traffic on the interface.
Total In Hit*	The number of hits from incoming traffic processed by all ACL entries (filters) in the ACL. A number is shown for each counter.
Out ACL	ID of the ACL used to filter the outgoing traffic on the interface.
Total Out Hit*	The number of hits from incoming traffic processed by all ACL entries (filters) in the ACL. A number is shown for each counter.
* The Total In Hit and Total Out Hit displays the total number of hits for all the ACL entries (or filters) in an ACL. For example, if an ACL has five entries and each entry processed matching conditions three times during the last minute, then the total Hits for the 1m counter is 15.	

Syntax: show access-list accounting brief [l2 | policy-based-routing | rate-limit]

The **l2** parameter limits the display to Layer 2 ACL accounting information.

The **policy-based-routing** parameter limits the display to policy based routing accounting information.

The **rate-limit** parameter limits the display to rate limiting ACL accounting information.

IPv4 ACL accounting statistics are displayed if no option is specified.

Displaying Statistics for an Interface

To display statistics for an interface, enter commands such as the following:

```
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.
Outbound/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>] in [l2 | policy-based-routing | rate-limit]

Use **ethernet** <slot>/<port> to display a report for a physical interface.

Use **ve** <ve-number> to display a report for the ports that are included in a virtual routing interface. For example, if ports 1/2, 1/4, and 1/6 are all members of ve 2, the report includes information for all three ports.

Use the **in** parameter to display statistics for incoming traffic; **out** for outgoing traffic.

The **l2** parameter limits the display to Layer 2 ACL accounting information.

The **policy-based-routing** parameter limits the display to policy based routing accounting information. This option is only available for incoming traffic.

The **rate-limit** parameter limits the display to rate limiting ACL accounting information.

Clearing the ACL Statistics

Statistics on the ACL account report can be cleared:

- When a software reload occurs
- When the ACL is bound to or unbound from an interface
- When you enter the **clear access-list** command, as in the following example:

```
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.

See “Extended ACL Syntax” on page 21-7 for the complete syntax for extended 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.

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 BigIron RX 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 BigIron RX to filter switched traffic within a virtual routing interface. When you enable the filtering, the BigIron RX 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

NOTE: The **ve-traffic** command is applied to a physical port. If you configure this command on a virtual routing interface that is a member of a tagged VLAN and ACLs are applied to that tagged VLAN, the ve-traffic command will not see the ACLs. The traffic will not be filtered. Ensure that the virtual routing interface is does not belong to a tagged VLAN.

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. See Table 21.2 on page 21-30 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. See Table 21.2 on page 21-30 for valid values.

Table 21.2: ICMP Message Types and Codes

ICMP Message Type	Type	Code
administratively-prohibited	3	13
any-icmp-type	x	x
destination-host-prohibited	3	10
destination-host-unknown	3	7
destination-net-prohibited	3	9
destination-network-unknown	3	6
echo	8	0
echo-reply	0	0
general-parameter-problem **Note:** This message type indicates that required option is missing.	12	1
host-precedence-violation	3	14
host-redirect	5	1
host-tos-redirect	5	3
host-tos-unreachable	3	12
host-unreachable	3	1
information-request	15	0

Table 21.2: ICMP Message Types and Codes

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 BigIron RX 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 22
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.

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 BigIron RX 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. See the section “Access Control List” on page 21-1 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).

NOTE: If you are configuring the ACL for use in a route map, always specify **permit**. Otherwise, the BigIron RX will ignore deny clauses and packets that match deny clauses are routed normally.

The <source-ip> parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE: To specify the host name instead of the IP address, the host name must be configured using the Foundry 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] route-map <map-name> permit | deny <num>

The <map-name> is a string of characters that names the map. Map names can be up to 32 characters in length. You can define an unlimited number of route maps on the BigIron RX, as long as system memory is available.

The **permit | deny** parameter specifies the action the BigIron RX will take if a route matches a match statement.

- If you specify **deny**, the BigIron RX 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 BigIron RX 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

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 | deny <num>`

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 BigIron RX 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 BigIron RX 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 BigIron RX to restore any PBR policies that were originally configured on the secondary ports.

Chapter 23
Configuring IP Multicast Traffic Reduction

BigIron RX forwards all IP multicast traffic by default based on the Layer 2 information in the packets. Optionally, you can enable the BigIron RX to make forwarding decisions in hardware, based on multicast group by enabling the IP Multicast Traffic Reduction feature.

When this feature is enabled, the BigIron RX 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 doesn't 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
```

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 must can be disabled globally by entering the **no ip multicast** command.

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.

- Active – When active IGMP mode is enabled, a Foundry device actively sends out IGMP queries to identify IP multicast groups on the network and makes entries in the IGMP table based on the Group Membership reports received from the network.

  NOTE: Routers in the network generally handle this operation. Use the active IGMP mode only when the device is in a stand-alone Layer 2 Switched network with no external IP multicast router attachments. In this case, enable the active IGMP mode on only one of the devices and leave the other devices configured for passive IGMP mode.

- Passive – When passive IGMP mode is enabled, the device listens for IGMP Group Membership reports but does not send IGMP queries. The passive mode is sometimes called "IGMP snooping". Use this mode when another device in the network is actively sending queries.

To enable active IGMP, enter the following command:

```
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 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 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 140 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 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

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.

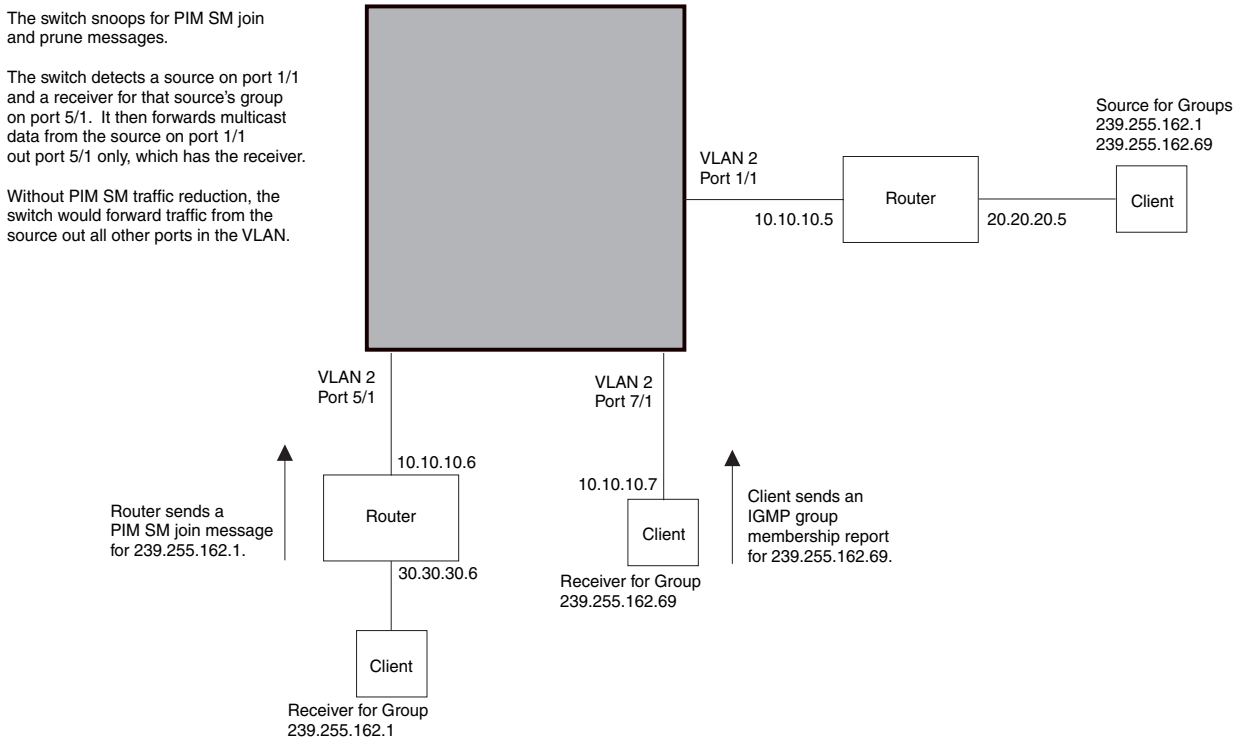
PIM SM traffic snooping requires IP multicast traffic reduction to be enabled on the device. IP multicast traffic reduction configures the device to listen for IGMP messages. PIM SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for 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 23.1 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 23.1 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.

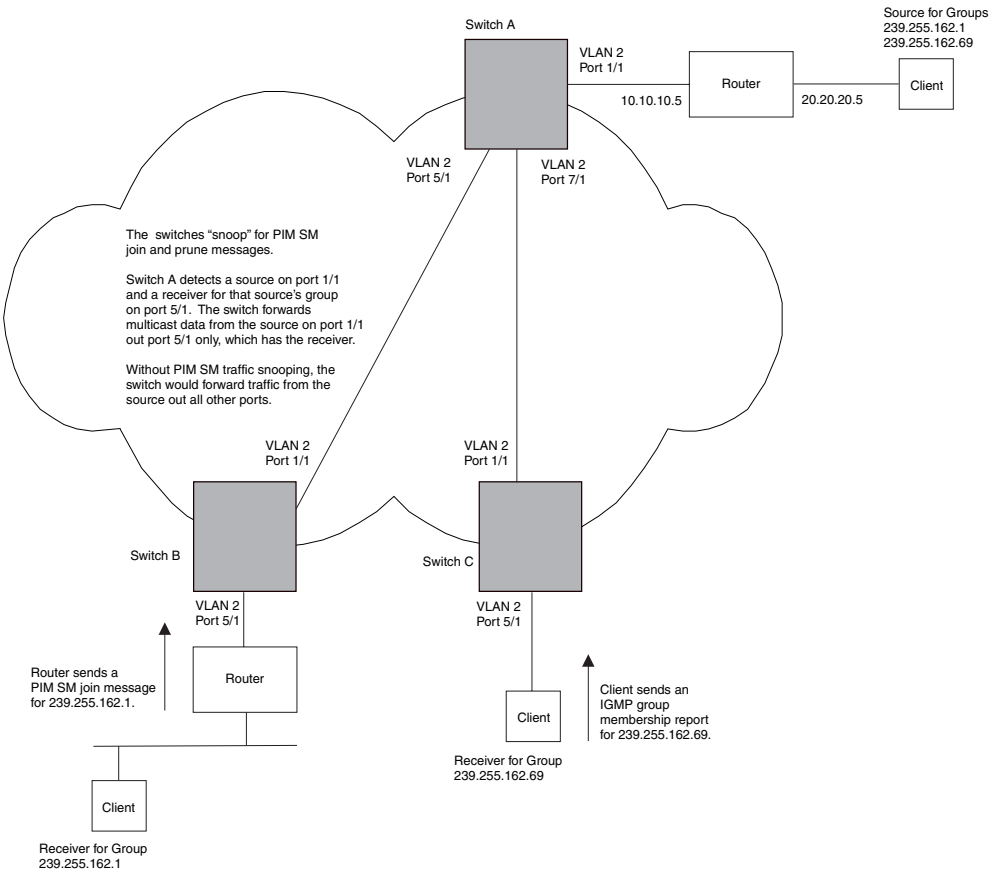
The IP multicast traffic reduction feature and the PIM SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM SM groups learned through join messages as well as MAC addresses learned through IGMP group membership reports. In this case, even 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 23.1.

Figure 23.2 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 23.2 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 BigIron RX, 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
```

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 igmp-snooping

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 Foundry 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 port(s) 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.

You also can display PIM SM information by entering the following command at any level of the CLI:

```
BigIron RX(config)# show ip multicast pimsm-snooping
PIMSM snooping is enabled

VLAN ID 100
  PIMSM neighbour list:
          31.31.31.4 : 12/2 expires 142 s
         31.31.31.13 : 10/7 expires 136 s
          31.31.31.2 : 3/1 expires 172 s
Number of Multicast Groups: 2
1    Group: 239.255.162.4 Num SG 4
     Forwarding ports : 3/1 12/2
     PIMv2 *G join ports : 3/1 12/2
     1     Source: (165.165.165.165, 10/7) FID 0x0bb3
           SG join ports: 12/2 10/7
     2     Source: (161.161.161.161, 10/7) FID 0x0bb2
           SG join ports: 12/2 3/1
     3     Source: (158.158.158.158, 10/7) FID 0x0bb1
           SG join ports: 12/2 3/1
     4     Source: (170.170.170.170, 10/7) FID 0x0baf
           SG join ports: 3/1 10/7
           (S, G) age 0 s
2    Group: 239.255.163.2 Num SG 1
     Forwarding ports : 10/7 12/2
     PIMv2 *G join ports : 10/7 12/2

     1     Source: (165.165.165.165, 3/1) FID 0x0bb5
            SG join ports: 12/2 10/7
```

Syntax: show ip multicast pimsm-snooping

This display shows the following information.

This Field...	Displays...
The PIM SM traffic snooping state	The first line of the display indicates whether the feature is enabled or disabled; and if it is enabled, if it is passive or active. The PIM SM traffic snooping feature requires the IP multicast traffic reduction feature.
VLAN ID	The port-based VLAN to which the neighbors and groups listed below the VLAN ID apply. Each port-based VLAN is a separate Layer 2 broadcast domain. **Note**: PIM SM traffic snooping requires the source and the receivers to be in the same port-based VLAN on the device. If the source and receivers are in different port-based VLANs, the device blocks the multicast traffic.
PIM SM Neighbor list	The PIM SM routers that are attached to the device's ports in the VLAN. The value following "expires" indicates how many seconds the device will wait for a hello message from the neighbor before determining that the neighbor is no longer present and removing the neighbor from the list.

This Field...	Displays...
Number of Multicast Group	The total number of groups for which the VLAN's ports have received PIM join or prune messages and IGMP group membership reports.
Multicast Group	The IP address of the multicast group. The "Num SG" entry indicates how many Source to Group flows are created for that Multicast Group as there can be more than one source for a given group. **Note**: The fid and camindex values are used by Foundry Technical Support for troubleshooting.
Forwarding Port	The port(s) attached to the group's receivers. A port is listed here when it receives a join message for the group, an IGMP membership report for the group, or both.
PIMv2 Group Port	The port(s) on which the BigIron RX has received PIM SM join messages for the group.
Source, Port list	The IP address of each PIM SM source and the BigIron RX ports connected to the receivers of the source.
SG join ports:	Ports from which a join message was received. The BigIron RX forwards the traffic only on this port.
(S, G) age	The actual aging value. If this entry shows the value 0 seconds, software age value is still 0 and the flow is programmed in the CAM. If the entry shows a value other than 0 seconds, then the CAM entry has aged out and the software aging has begun. Once this age value reaches the Group Age value the entry will be deleted from the table. Group age value can be can be from 10 – 1220 seconds. The default is 140 seconds.

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 24
Configuring IP Multicast Protocols

This chapter describes how to configure Foundry BigIron RX for the following IP multicast protocol and versions:

- Internet Group Management Protocol (IGMP) V1 and V2
- Protocol Independent Multicast Dense mode (PIM DM) V1 (draft-ietf-pim-dm-05) and V2 (draft-ietf-pim-v2-dm-03)
- PIM Sparse mode (PIM SM) V2 (RFC 2362)
- Distance Vector Multicast Routing Protocol (DVMRP) V2 (RFC 1075)

NOTE: Each 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.

NOTE: This chapter applies only to IP multicast routing. To configure Layer 2 IP multicast features, see the "Configuring IP Multicast Traffic Reduction" on page 23-1 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.

BigIron RX 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 BigIron RX. Also, the CAM can hold up to 1535 IPv4 multicast entries.

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.

Changing Global IP Multicast Parameters

The sections below apply to PIM-DM, PIM-SM, and DVMRP.

Defining the Maximum Number of Multicast Flows

The Multicast Flow table is shared by PIM and DVMRP. It defines the maximum number of flows for a PIM or DVMRP multicast switching that can be written in hardware (CAM). To define the maximum number of entries for the Multicast Flow table, enter a command such as the following:

```
BigIron RX(config)# system-max multicast-flow 2048
```

Syntax: system-max multicast-flow <num>

The <num> parameter specifies the maximum number of PIM and DVMRP multicast cache flows that can be stored in the CAM. Enter a number from 512 – 2048. The default is 1024.

NOTE: Do not set this maximum too high since you may run out of resources in the CAM.

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.

Changing IGMP V1 and V2 Parameters

IGMP allows Foundry 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, inserts the group information in an IGMP packet, and forwards the packet to IP Multicast neighbors.

The following IGMP V1 and V2 parameters apply to PIM and DVMRP:

- IGMP query interval – Specifies how often the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 Foundry 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**

PIM Dense

NOTE: This section describes the “dense” mode of PIM, described in RFC 1075. See “PIM Sparse” on page 24-12 for information about PIM Sparse.

PIM was introduced to simplify some of the complexity of the routing protocol at the cost of additional overhead tied with a greater replication of forwarded multicast packets. PIM is similar to DVMRP in that PIM builds source-routed multicast delivery trees and employs reverse path check when forwarding multicast packets.

There are two modes in which PIM operates: Dense and Sparse. The Dense Mode is suitable for densely populated multicast groups, primarily in the LAN environment. The Sparse Mode is suitable for sparsely populated multicast groups with the focus on WAN.

PIM primarily differs from DVMRP by using the IP routing table instead of maintaining its own, thereby being routing protocol independent.

Initiating PIM Multicasts on a Network

Once PIM is enabled on each router, a network user can begin a video conference multicast from the server on R1 as shown in Figure 24.1. 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 24.1, 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 24.1 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 24.2, 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 24.2. 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 24.1 Transmission of multicast packets from the source to host group members

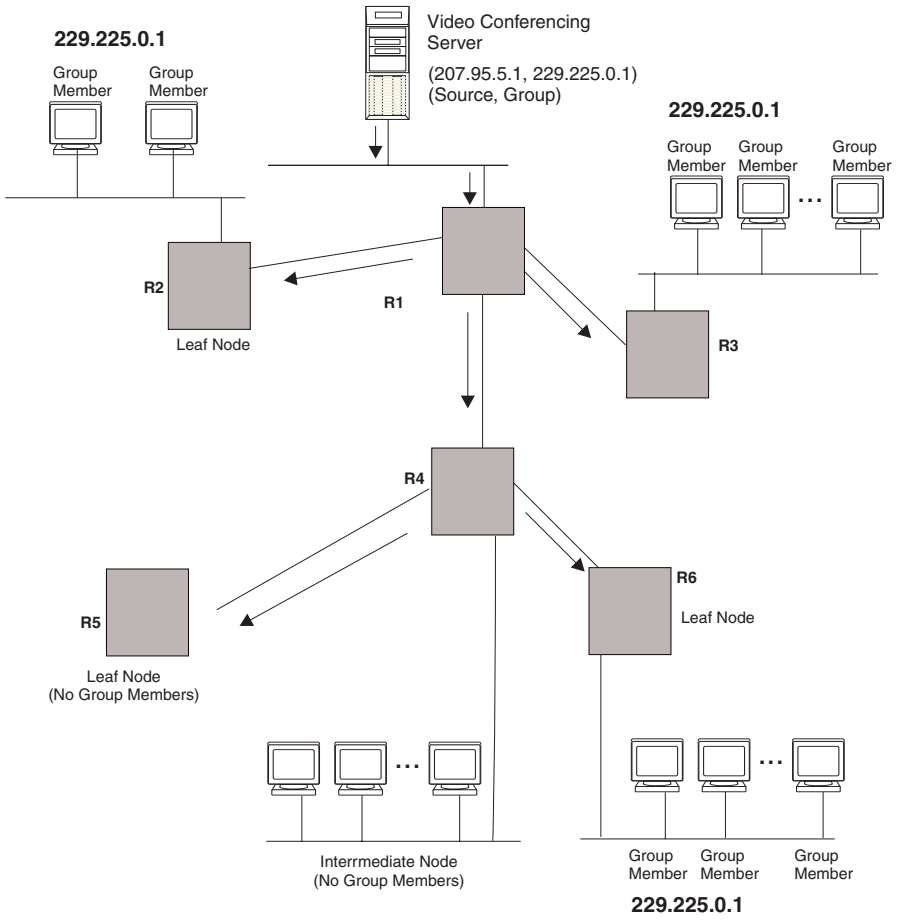
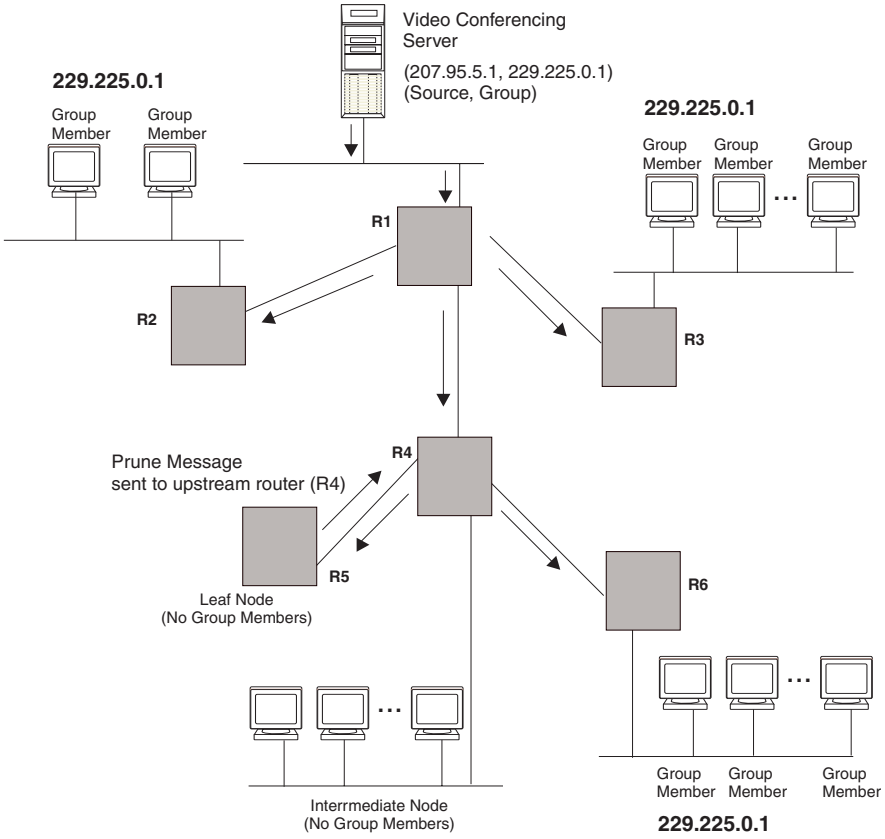


Figure 24.2 Pruning leaf nodes from a multicast tree



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 BigIron RX 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 doesn't 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 BigIron RX running PIM to a device that is running PIM V1, you must change the PIM version on the BigIron RX 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. See "Configuring PIM Sparse" on page 24-13 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 Foundry routers that connect the various buildings need to be configured to support PIM multicasts from the designated video conference server as shown in Figure 24.1 on page 24-6.

PIM is enabled on each of the Foundry routers shown in Figure 24.1, 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

NOTE: When PIM routing is enabled, the line rate for receive traffic is reduced by about 5%. The reduction occurs due to overhead from the VLAN multicasting feature, which PIM routing uses. This behavior is normal and does not indicate a problem with the device.

The 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 BigIron RX (**router pim** level) only.

Globally Enabling and Disabling PIM without Deleting Multicast Configuration

As stated above entering a **no router pim** command deletes PIM configuration. If you want to disable PIM without deleting any PIM configuration, enter the following command:

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# disable-pim
```

Syntax: [no] disable-pim

Use the [no] version of the command to re-enable PIM.

Enabling a PIM version

To enable PIM on an interface, globally enable PIM, then enable PIM on interface 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 Foundry 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> can be 0 - 3 seconds. A value of 0 causes the PIM router to stop traffic immediately upon receiving a prune message. The default is 3 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 <10-3600>

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>

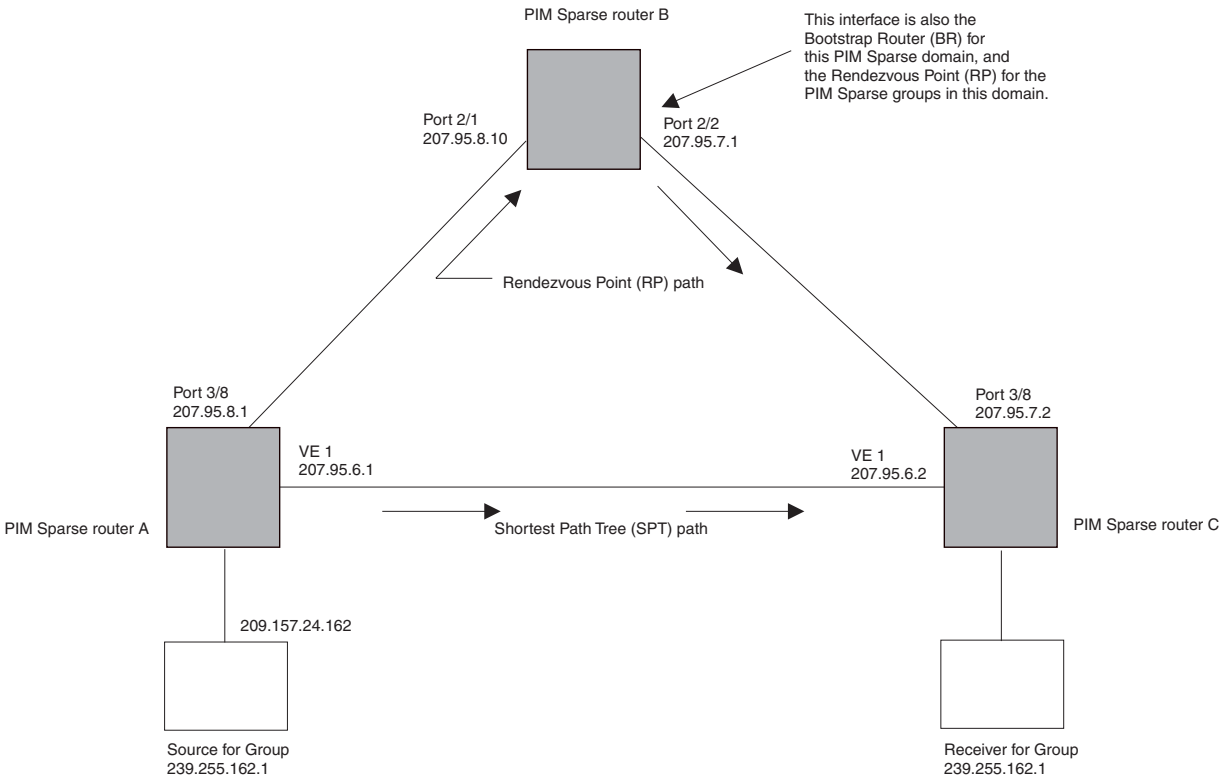
PIM Sparse

The BigIron RX supports Protocol Independent Multicast (PIM) Sparse version 2. PIM Sparse provides multicasting that is especially suitable for widely distributed multicast environments. The Foundry 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 24.3 shows a simple example of a PIM Sparse domain. This example shows three BigIron RX devices configured as PIM Sparse routers. The configuration is described in detail following the figure.

Figure 24.3 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 Foundry 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 24.3, PIM Sparse router B is the BSR. Port 2/2 is configured as a candidate BSR.
- RP – The RP is the meeting point for PIM Sparse sources and receivers. A PIM Sparse domain can have multiple RPs, but each PIM Sparse multicast group address can have only one active RP. PIM Sparse 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 24.3, PIM Sparse router B is the RP. Port 2/2 is configured as a candidate Rendezvous Point (RP).

  To enhance overall network performance, Foundry 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: Foundry Networks recommends that you configure the same ports as candidate BSRs and RPs.

RP Paths and SPT Paths

Figure 24.3 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 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 24.3, the BigIron RX 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.

Configuring PIM Sparse

To configure a BigIron RX 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 Foundry 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: Foundry Networks recommends that you configure the same BigIron RX as both the BSR and the RP.

Current Limitations

The implementation of PIM Sparse in the current software release has the following limitations:

- PIM Border Routers (PMBRs) are not supported. Thus, you cannot configure a Foundry routing interface as a PMBR interface for PIM Sparse.
- 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 BigIron RX, 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 BigIron RX 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 BigIron RX as a candidate PIM Sparse Bootstrap Router (BSR) and candidate Rendezvous Point (RP). You can configure a BigIron RX as a PIM Sparse router without configuring the BigIron RX as a candidate BSR and RP. However, if you do configure the BigIron RX as one of these, Foundry Networks recommends that you configure the BigIron RX as both of these. See “Configuring BSRs” on page 24-15.

Entering a **[no] router pim** command does the following:

- Disables PIM or DVMRP.
- Removes all configuration for PIM multicast on a BigIron RX (**router pim** level) only.

Globally Enabling and Disabling PIM without Deleting Multicast Configuration

As stated above entering a **no router pim** command deletes PIM configuration. If you want to disable PIM without deleting any PIM configuration, enter the following command:

```
BigIron RX(config)# router pim
BigIron RX(config-pim-router)# disable-pim
```

Syntax: [no] disable-pim

Use the [no] version of the command to re-enable PIM.

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 Foundry 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 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 Foundry Networks recommends that you configure the same interface on the same BigIron RX as both a BSR and an RP.

This section presents how to configure BSRs. Refer to “Configuring RPs” on page 24-15 for instructions on how to configure RPs.

To configure the BigIron RX 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 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.

NOTE: Foundry Networks recommends you specify 30 for IP version 4 (IPv4) networks.

The <priority> specifies the BSR priority. You can specify a value from 0 – 255. When the election process for BSR takes place, the candidate BSR with the highest priority becomes the BSR. The default is 0.

Configuring RPs

Enter a command such as the following to configure the BigIron RX 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 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.

By default, this command configures the BigIron RX as a candidate RP for all group numbers beginning with 224. As a result, the BigIron RX 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 BigIron RX 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 BigIron RX is a candidate RP for all groups that begin with 224.126. When you add a range, you override the default. The BigIron RX then becomes a candidate RP only for the group address range(s) you add.

You also can change the group numbers for which the BigIron RX 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 BigIron RX 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

Foundry Networks 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 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 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 BigIron RX will use the specified RP and ignore group-to-RP mappings received from the BSR.

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 BigIron RX 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 BigIron RX 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 BigIron RX itself as the root of the tree. The first time a BigIron RX is configured as a PIM router receives a packet for a PIM receiver, the BigIron RX sends the packet to the RP for the group. The BigIron RX also calculates the SPT from itself to the receiver. The next time the BigIron RX receives a PIM Sparse packet for the receiver, the BigIron RX 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 BigIron RX maintains a separate counter for each PIM Sparse source-group pair.

After the BigIron RX receives a packet for a given source-group pair, the BigIron RX starts a PIM data timer for that source-group pair. If the BigIron RX 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 BigIron RX receives a packet for the source-group pair.

You can change the number of packets that the BigIron RX sends using the RP before switching to using the SPT.

To change the number of packets the BigIron RX 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 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.

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)# 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.

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 24.3.

This display shows the following information.

This Field...	Displays...
Global PIM Sparse mode settings	
Hello interval	How frequently the BigIron RX 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 BigIron RX 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 BigIron RX 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 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 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 a PIM Sparse group. When forwarding packets from 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. You can change the Join/Prune interval if needed. See "Changing the PIM Join and Prune Message Interval" on page 24-17.
SPT Threshold	The number of packets the BigIron RX 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.

This Field...	Displays...
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 BigIron RX is forwarding. **Note**: This list can include groups that are not PIM Sparse groups. If interfaces on the BigIron RX 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 BigIron RX 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 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(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 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.
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 BigIron RX can be a BSR. The default is 32 bits, which allows the BigIron RX to be a BSR for any valid IP 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.

This Field...	Displays...
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 IP 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(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 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(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 BigIron RX is a candidate RP.
RP	Indicates the IP address of the Rendezvous Point (RP). **Note**: This field appears only if this BigIron RX 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 BigIron RX is a candidate RP.

This Field...	Displays...
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(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.

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, via 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.

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 BigIron RX learned the identity of the RP.

This Field...	Displays...
Info source	Indicates the IP address on which the RP information was received. Following the IP 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(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.

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 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 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.

This Field...	Displays...
Port	The interface through which the BigIron RX is connected to the neighbor.
Neighbor	The IP interface of the PIM neighbor interface.
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 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 via 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 **sh ip pim mcache** to view information about the upstream neighbor.

The following example outputs show other messages that the BigIron RX 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.

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 BigIron RX 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 BigIron RX 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.

This Field...	Displays...
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 BigIron RX 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 BigIron RX 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 BigIron RX, statistics for these interfaces are listed first by the display.

This display shows the following information.

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.

This Field...	Displays...
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.

DVMRP Overview

The BigIron RX 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 24.4. 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 24.4, 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 24.5, 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 24.4 Downstream broadcast of IP multicast packets from source host

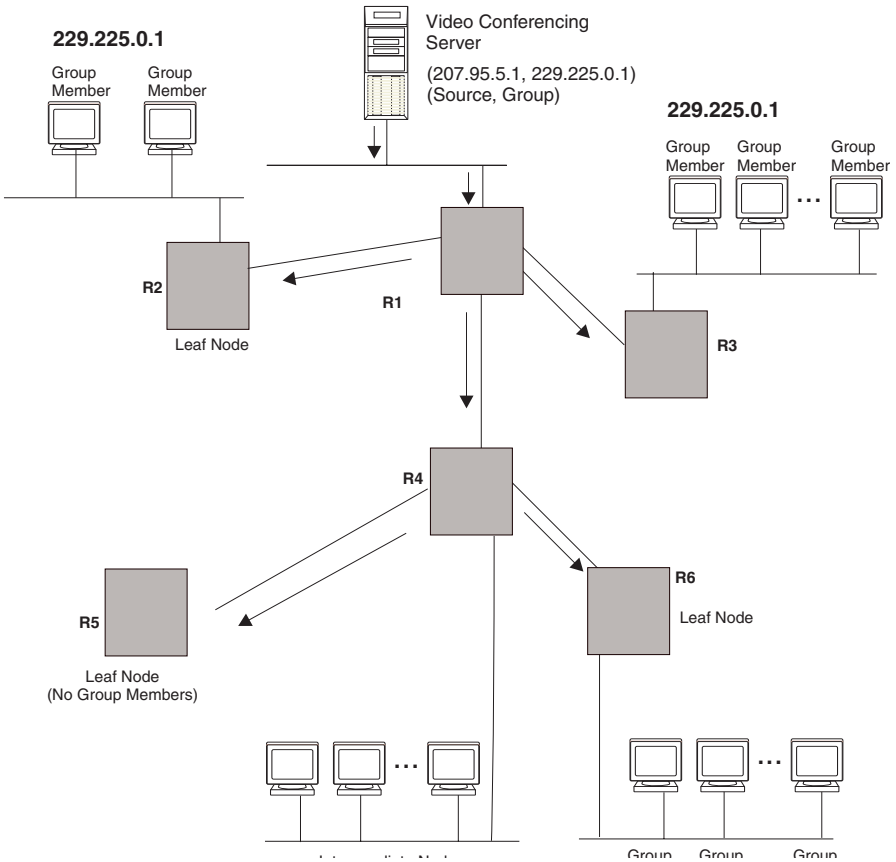
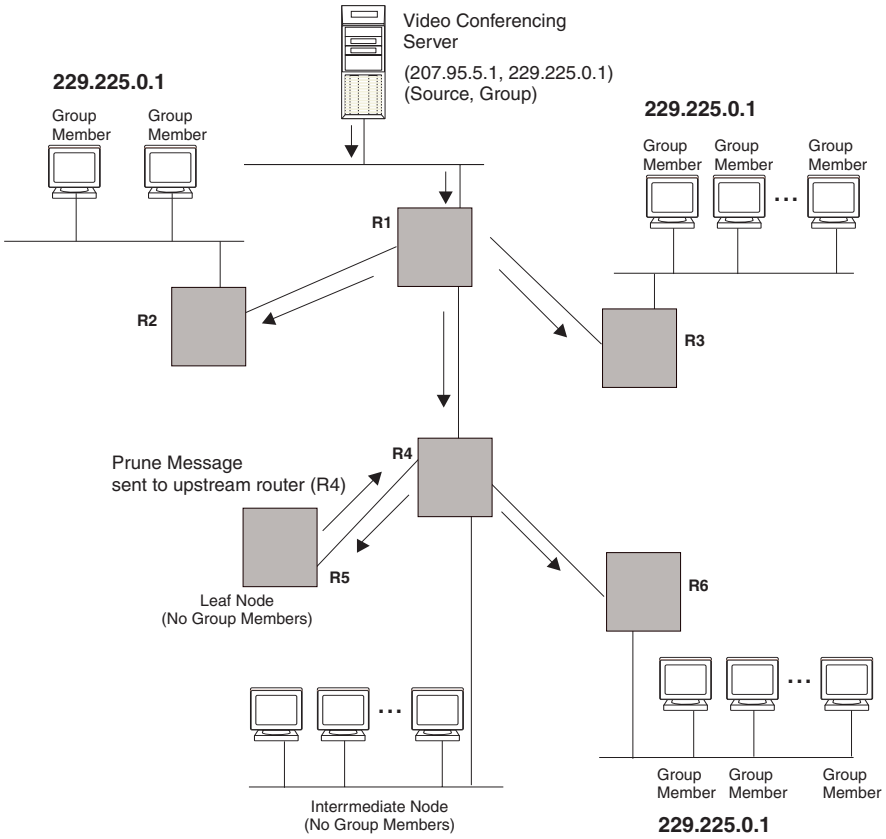


Figure 24.5 Pruning leaf nodes from a multicast tree



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 BigIron RXes that connect the various buildings need to be configured to support DVMRP multicasts from the designated video conference server as seen in Figure 24.4.

DVMRP is enabled on each of the BigIron RX devices, shown in Figure 24.4, on which multicasts are expected. You can enable DVMRP on each BigIron RX independently or remotely from one BigIron RX 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 BigIron RX (**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 BigIron RX, 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: 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 3/5
BigIron RX(config-if-e10000-3/5)# ip dvmrp metric 15
```

Syntax: 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: 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 Foundry device can display with this command.

```
BigIron RX# show ip dvmrp rpf 1.1.20.2|
directly connected or via 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

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 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 as shown in the examples below.

To add static routes to multicast router A (see Figure 24.6), enter commands such as the following:

```
PIMRouterA(config)# ip mroute 1 207.95.10.0 255.255.255.0 interface ethernet 1/2
distance 1
PIMRouterA(config)# ip mroute 2 0.0.0.0 0.0.0.0 interface ethernet 2/3 distance 1
PIMRouterA(config)# write memory
```

Syntax: mroute <route-num> <ip-addr> interface ethernet <slot>/<portnum> | ve <num> [distance <num>]

Or

Syntax: mroute <route-num> <ip-addr> rpf_address <rpf-num>

The <route-num> parameter specifies the route number.

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 BigIron RX prefers the path with the lower administrative distance.

NOTE: Regardless of the administrative distances, the BigIron RX always prefers directly connected routes over other routes.

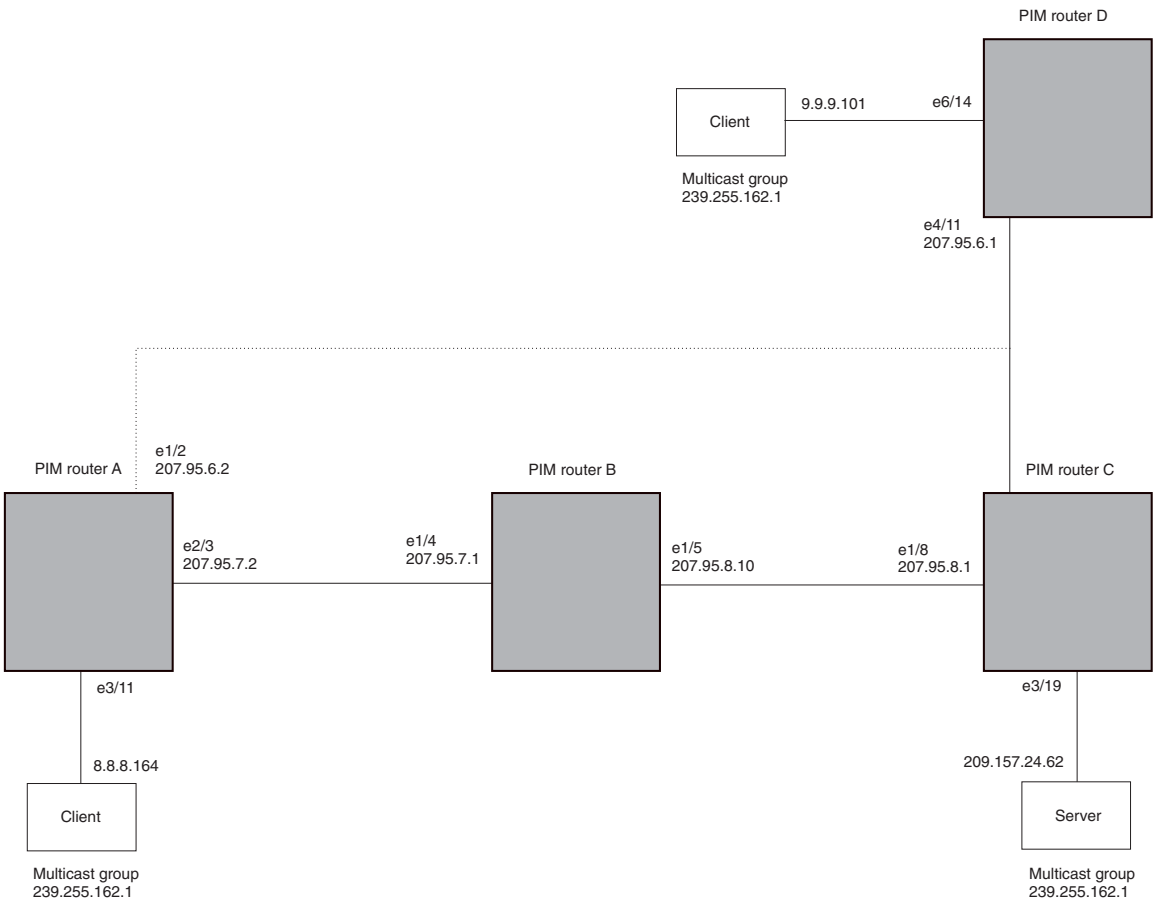
The **rpf_address** <rpf-num> parameter specifies an RPF number.

The example above configures two static multicast routes. The first route is for a specific source network, 207.95.10.0/24. If the BigIron RX 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 24.6 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 BigIron RX uses the default static route, assign a low administrative distance value. When comparing multiple paths for a route, the BigIron RX prefers the path with the lower administrative distance.

Figure 24.6 Example multicast static routes



To add a static route to a virtual interface, enter commands such as the following:

```
BigIron RX(config)# mroute 3 0.0.0.0 0.0.0.0 int ve 1 distance 1
BigIron RX(config)# write memory
```


Chapter 25
Configuring 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 BigIron RX 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 BigIron RX receives a RIP update from another router that contains a path with fewer hops than the path stored in the BigIron RX's route table, the BigIron RX replaces the older route with the newer one. The BigIron RX then includes the new path in the updates it sends to other RIP routers, including BigIron RX

RIP routers, including the BigIron RX, 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.

A BigIron RX supports the following RIP versions:

- Version 1
- V1 compatible with V2
- Version 2 (the default)

RIP Parameters and Defaults

The following tables list the RIP parameters, their default values, and where to find configuration information.

RIP Global Parameters

Table 25.1 lists the global RIP parameters and their default values, and indicates where you can find configuration information.

Table 25.1: RIP Global Parameters

Parameter	Description	Default	See page...
RIP state	The global state of the protocol **Note**: You also must enable the protocol on individual interfaces. Globally enabling the protocol does not allow interfaces to send and receive RIP information. See Table 25.2 on page 25-3.	Disabled	25-3
Administrative distance	The administrative distance is a numeric value assigned to each type of route on the router. When the router is selecting from among multiple routes (sometimes of different origins) to the same destination, the router compares the administrative distances of the routes and selects the route with the lowest administrative distance.	120	25-4
Redistribution	RIP can redistribute routes from other routing protocols such as OSPF and BGP4 into RIP. A redistributed route is one that a router learns through another protocol, then distributes into RIP.	Disabled	25-4
Redistribution metric	RIP assigns a RIP metric (cost) to each external route redistributed from another routing protocol into RIP. This parameter applies to routes that are redistributed from other protocols into RIP.	1 (one)	25-5
Learning default routes	The router can learn default routes from its RIP neighbors. **Note**: You also can enable or disable this parameter on an individual interface basis. See Table 25.2 on page 25-3.	Disabled	25-6
Advertising and learning with specific neighbors	The BigIron RX learns and advertises RIP routes with all its neighbors by default. You can prevent the BigIron RX from advertising routes to specific neighbors or learning routes from specific neighbors.	Learning and advertising permitted for all neighbors	25-6

RIP Interface Parameters

Table 25.2 lists the interface-level RIP parameters and their default values, and indicates where you can find configuration information.

Table 25.2: RIP Interface Parameters

Parameter	Description	Default	See page...
RIP state and version	The state of the protocol and the version that is supported on the interface. The version can be one of the following: • Version 1 only • Version 2 only • Version 1, but also compatible with version 2 **Note**: You also must enable RIP globally.	Disabled	25-3
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	25-4
Learning default routes	Locally overrides the global setting. See Table 25.1 on page 25-2.	Disabled	25-6
Loop prevention	The method a router uses to prevent routing loops caused by advertising a route on the same interface as the one on which the router learned the route. • Split horizon – The router does not advertise a route on the same interface as the one on which the router learned the route. • Poison reverse – The router assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which the router learned the route.	Split Horizon **Note**: Disabling poison reverse enables split horizon on the interface.	25-6
Advertising and learning specific routes	You can control the routes that a BigIron RX learns or advertises.	The BigIron RX learns and advertises all RIP routes on all interfaces.	25-7

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 BigIron RX port increases the cost of a RIP route that is learned on the port. The BigIron RX 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 BigIron RX 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 BigIron RX assigns the default RIP administrative distance (120) to RIP routes. When comparing routes based on administrative distance, the BigIron RX selects the route with the lower distance. You can change the administrative distance for RIP routes.

NOTE: See "Changing Administrative Distances" on page 27-21 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 rediscrtibution 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 BigIron RX 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 BigIron RX assigns, up to 15.

To change the RIP metric the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX so that the device does not learn any RIP routes from any RIP neighbors.

The following commands configure the BigIron RX 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 BigIron RX does not advertise a route on the same interface as the one on which the router learned the route.
- Poison reverse – The BigIron RX 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 BigIron RX 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). See "Configuring VRRP and VRRPE" on page 15-1.

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 BigIron RX learns from its neighbor on the interface.

Out applies the prefix list to routes the BigIron RX 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 BigIron RX learns from its neighbor on the interface.

Out applies the route map to routes the BigIron RX 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 25.3: CLI Display of Neighbor Filter Information

This Field...	Displays...
RIP Summary area	Shows the current configuration of RIP on the device.
Statis metric	Shows the static metric configuration. ".not defined" means the route map has not been distributed.
OSPF metric	Shows what OSPF route map has been applied.
Neighbor Filter Table area	
Index	The filter number. You assign this number when you configure the filter.

Table 25.3: CLI Display of Neighbor Filter Information (Continued)

This Field...	Displays...
Action	The action the router takes for RIP route packets to or from the specified neighbor: • deny – If the filter is applied to an interface's outbound filter group, the filter prevents the router from advertising RIP routes to the specified neighbor on that interface. If the filter is applied to an interface's inbound filter group, the filter prevents the router from receiving RIP updates from the specified neighbor. • permit – If the filter is applied to an interface's outbound filter group, the filter allows the router to advertise RIP routes to the specified neighbor on that interface. If the filter is applied to an interface's inbound filter group, the filter allows the router to receive RIP updates from the specified neighbor.
Neighbor IP Address	The IP address of the RIP neighbor.

Chapter 26
Configuring OSPF Version 2 (IPv4)

This chapter describes how to configure OSPF Version 2 on a BigIron RX. OSPF Version 2 is supported on devices running IPv4.

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 BigIron RX 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 26.1 on page 26-2. 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.

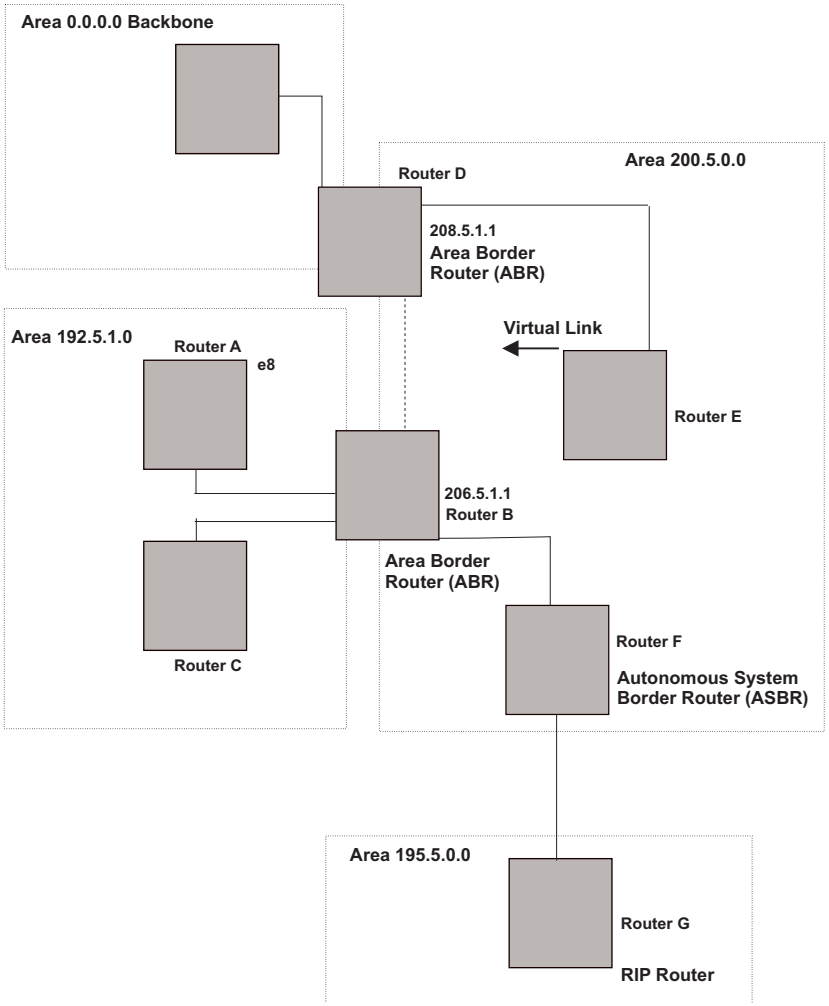
You can further limit the broadcast area of flooding by defining an area range. The area range allows you to assign an aggregate value to a range of IP addresses. This aggregate value becomes the address that is advertised instead all of the individual addresses it represents being advertised. You can 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, see “Enable Route Redistribution” on page 26-24.

Figure 26.1 OSPF operating in a network



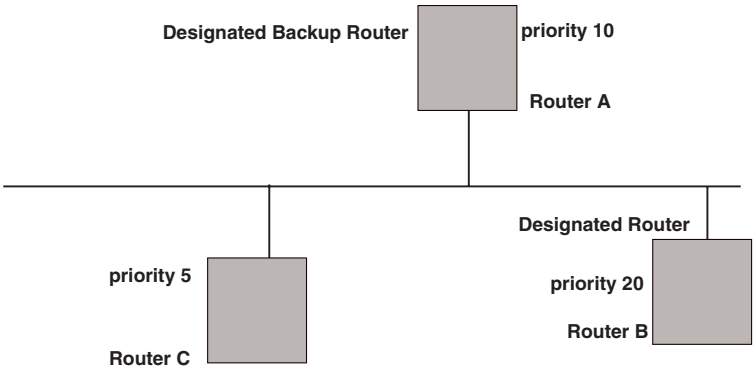
Designated Routers in Multi-Access Networks

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 26.2

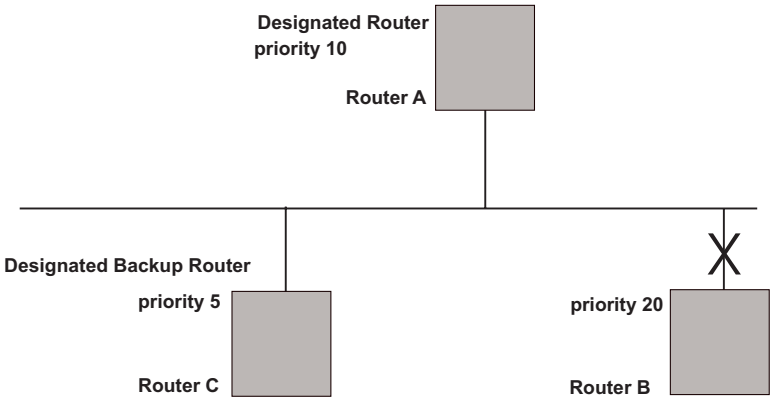
Figure 26.2 Designated and backup router election



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 26.3.

NOTE: Priority is a configurable option at the interface level. You can use this parameter to help bias one router as the DR.

Figure 26.3 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 Foundry 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, see “Changing the Router ID” on page 18-21.

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

Foundry routers are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification. Foundry 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, see "Modify OSPF Standard Compliance Setting" on page 26-31.

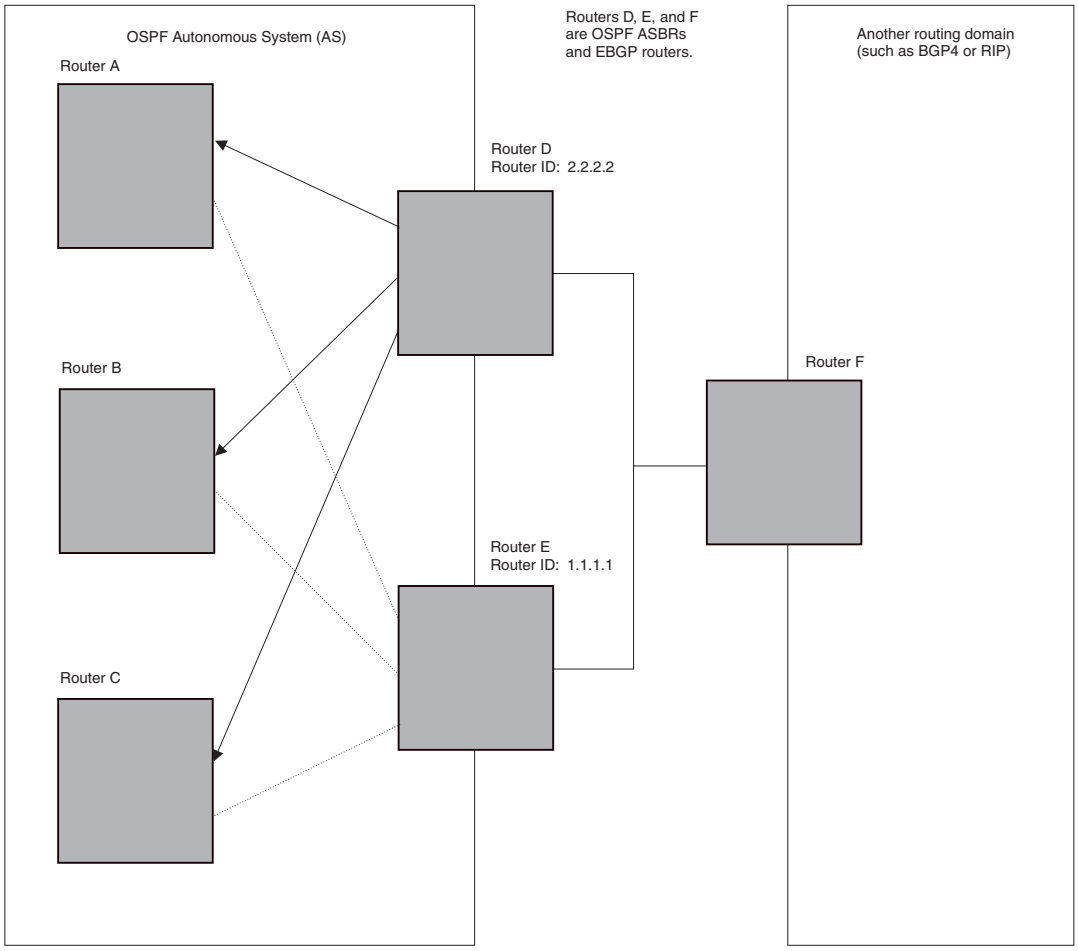
Reduction of Equivalent AS External LSAs

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 BigIron RX optimizes OSPF by eliminating duplicate AS External LSAs in this case. The BigIron RX 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 26.4 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 26.4 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 BigIron RX switches that flush the duplicate AS External LSAs have more memory for other OSPF data. In Figure 26.4, 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 26.4 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 10.0.0.255.

   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 BigIron RX 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 26.1 on page 26-2, 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 BigIron RX 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 BigIron RX 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 BigIron RX still accepts summary LSAs from OSPF neighbors and floods them to other neighbors. The BigIron RX 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 BigIron RX 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)

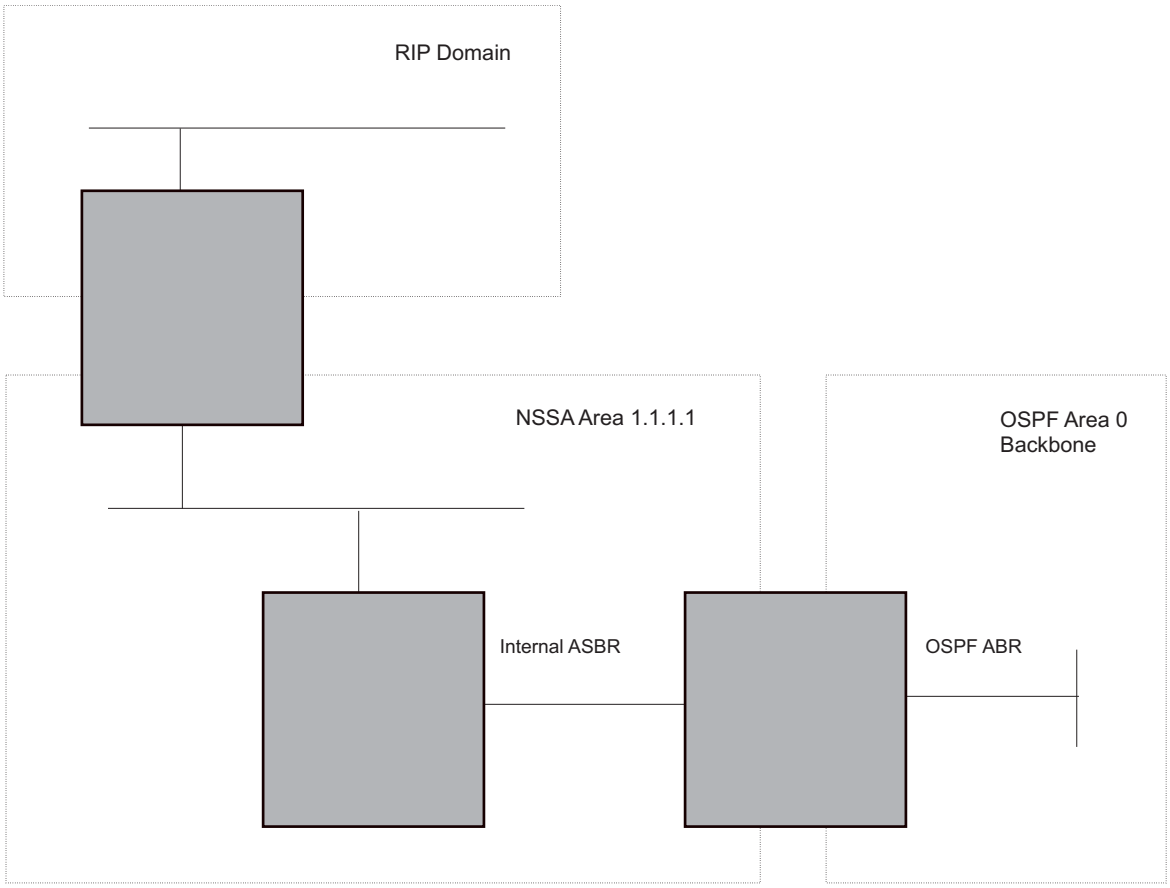
The OSPF Not So Stubby Area (NSSA) feature enables you to configure OSPF areas that provide the benefits of stub areas, but that also are capable of importing external route information. OSPF does not flood external routes from other areas into an NSSA, but does translate and flood route information 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 Foundry implementation of NSSA is based on RFC 1587.

Figure 26.5 shows an example of an OSPF network containing an NSSA.

Figure 26.5 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 LSA(s) 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 BigIron RX 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 BigIron RX 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. 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.
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. See "Assigning an IP Address to an Ethernet Port" on page 18-11.
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 optional **0 | 1** parameter with the **authentication-key** and **md5-authentication key-id** parameters affects encryption.

For added security, the BigIron RX 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 BigIron RX 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.

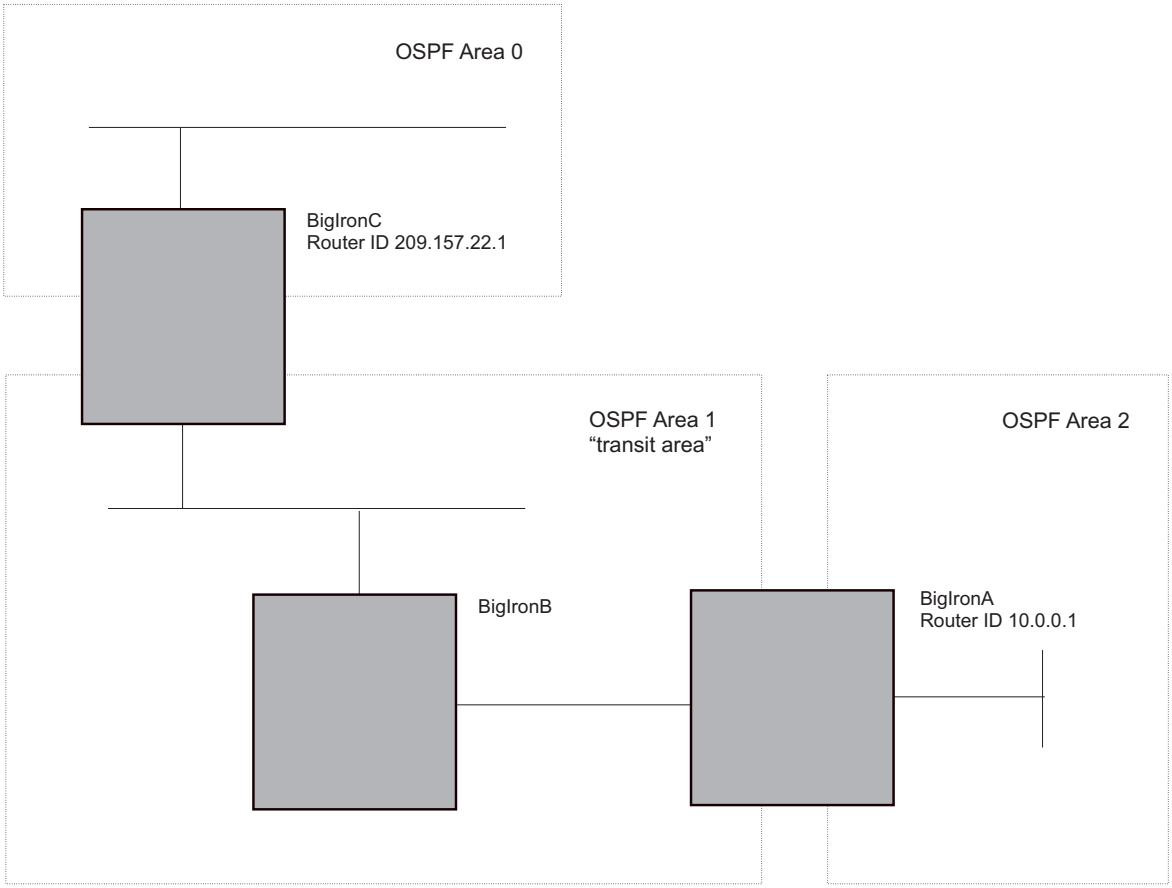
Two parameters fields must be defined for all virtual links—transit area ID and neighbor router.

- 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 Foundry 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, see “Changing the Router ID” on page 18-21.

NOTE: When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

Figure 26.6 Defining OSPF virtual links within a network



EXAMPLE:

Figure 26.6 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.

See “Modify Virtual Link Parameters” on page 26-19 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.
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.

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 BigIron RX advertises its interfaces and their costs to OSPF neighbors. For example, if an interface has an OSPF cost of ten, the BigIron RX 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 BigIron RX sends a link-state update to update the costs of interfaces advertised by the BigIron RX.

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

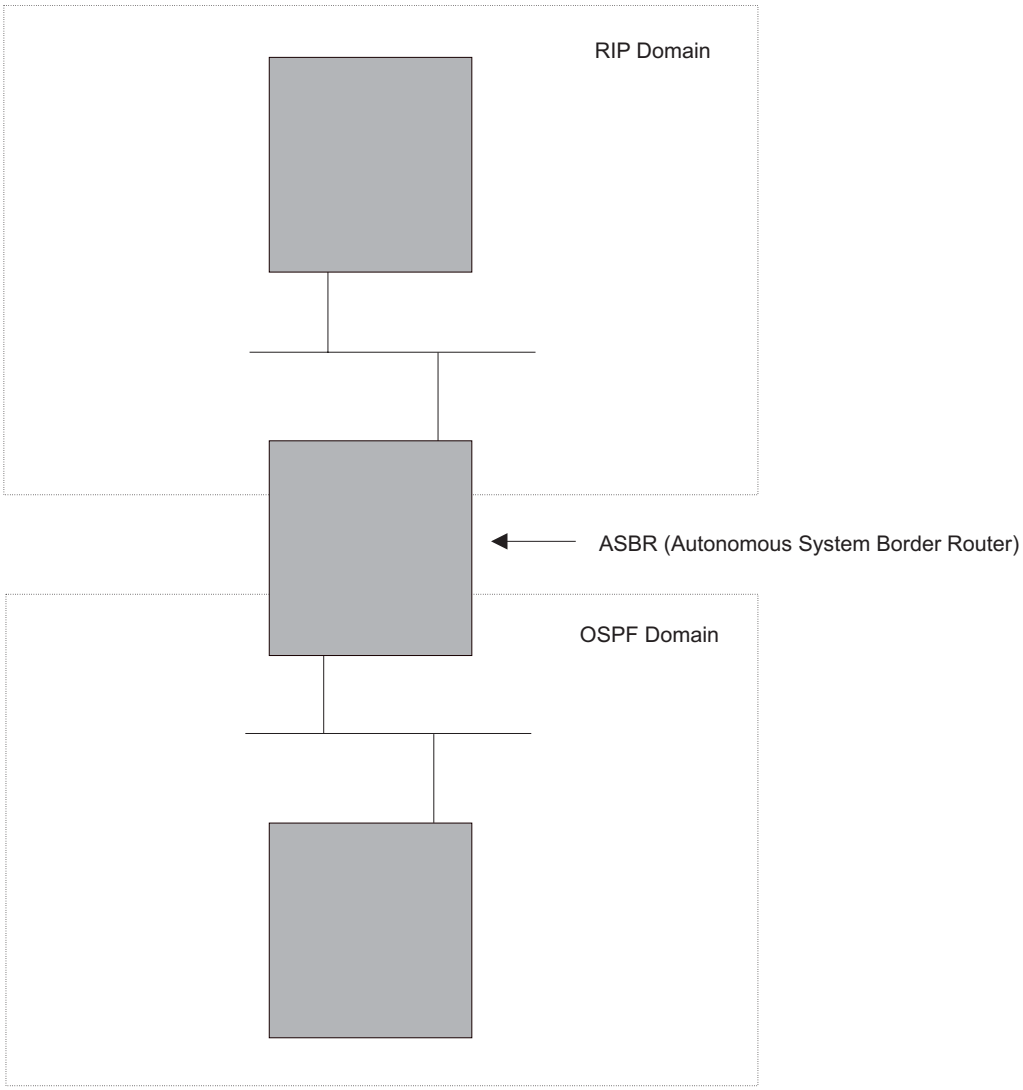
Route redistribution imports and translates different protocol routes into a specified protocol type. On the BigIron RX, redistribution is supported for static routes, ISIS, OSPF, RIP, and BGP4. OSPF redistribution supports the import of static, ISIS, RIP, and BGP4 routes into OSPF routes.

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 26.7 on page 26-23, an administrator wants to configure the BigIron RX 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 26.7 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 – 15.

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 – 15. The default is 10.

Enable Route Redistribution

NOTE: Do not enable redistribution until you have configured the redistribution route map. Otherwise, you might accidentally overload the network with routes you did not intend to redistribute.

To enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
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.

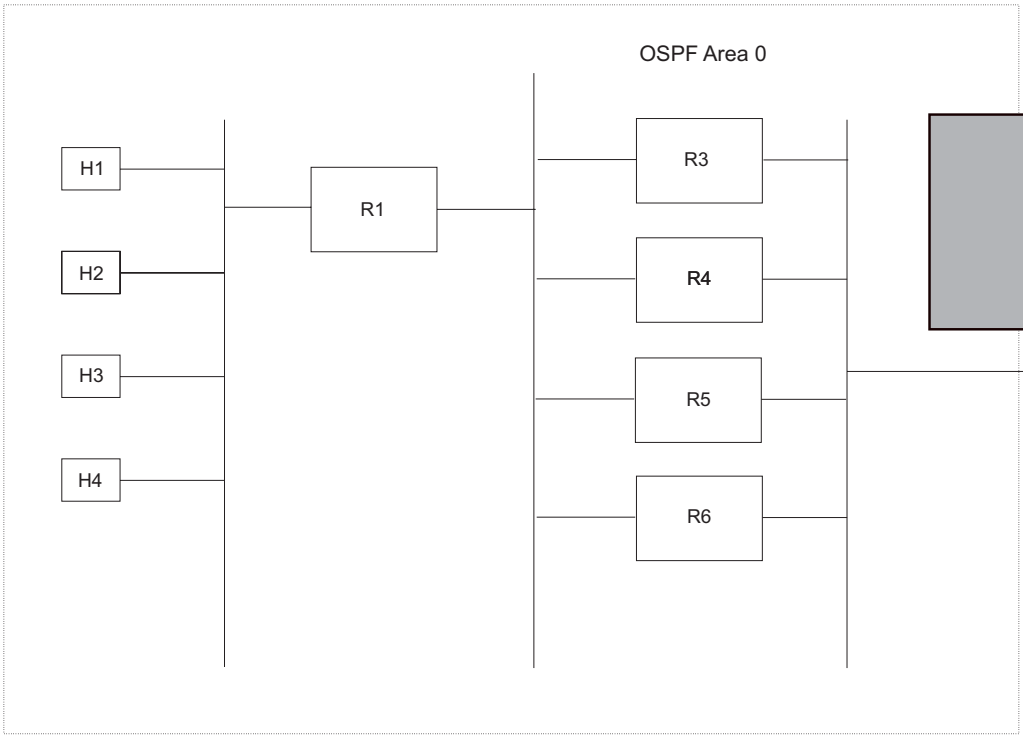
NOTE: For an external route that is redistributed into OSPF through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map. The **default-metric** <num> command has no effect on the route. This behavior is different from a 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 26.8 shows an example of an OSPF network containing multiple paths to a destination (in this case, R1).

Figure 26.8 Example OSPF network with four equal-cost paths



In the example in Figure 26.8, the BigIron RX has four paths to R1:

- BigIron RX ->R3
- BigIron RX ->R4
- BigIron RX ->R5
- BigIron RX ->R6

Normally, the BigIron RX 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 BigIron RX 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, see "Configuring IP Load Sharing" on page 18-38.

Configure External Route Summarization

When the BigIron RX 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 BigIron RX, no action is taken if the BigIron RX has already advertised the aggregate route; otherwise the BigIron RX advertises the aggregate route. If an imported route that falls with in a configured address range is removed by the BigIron RX, no action is taken if there are other imported route(s) that fall with in the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX does not advertise the default route into the OSPF domain. If you want the BigIron RX to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPF default route origination, the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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. See "Assign a Not-So-Stubby Area (NSSA)" on page 26-10.

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.

Modify SPF Timers

The BigIron RX uses the following timers when calculating the shortest path for OSPF routes:

- SPF delay – When the BigIron RX 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 BigIron RX waits for a specific amount of time between consecutive SPF calculations. By default, the BigIron RX 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 BigIron RX 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 BigIron RX 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. See “Changing Administrative Distances” on page 27-21 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 BigIron RX’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 BigIron RX has multiple routes for the same network from different protocols. The BigIron RX 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 BigIron RX 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 BigIron RX refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the BigIron RX 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 BigIron RX refreshes the group of accumulated LSAs and sends the group together in the same packet(s).

Usage Guidelines

The pacing interval is inversely proportional to the number of LSAs the BigIron RX 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
```

Modify OSPF Traps Generated

OSPF traps as defined by RFC 1850 are supported on BigIron RX.

You can disable all or specific OSPF trap generation 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>

Modify OSPF Standard Compliance Setting

The BigIron RX 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 BigIron RX 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 – see “Displaying General OSPF Configuration Information” on page 26-33.
- CPU utilization statistics – see “Displaying CPU Utilization and Other OSPF Tasks” on page 26-34.
- Area information – see “Displaying OSPF Area Information” on page 26-35.
- Neighbor information – see “Displaying OSPF Neighbor Information” on page 26-36.
- Interface information – see “Displaying OSPF Interface Information” on page 26-38.
- Route information – see “Displaying OSPF Route Information” on page 26-40.
- External link state information – see “Displaying OSPF External Link State Information” on page 26-42.
- Link state information – see “Displaying OSPF Database Link State Information” on page 26-43.
- Virtual Neighbor information – see “Displaying OSPF Virtual Neighbor and Link Information” on page 26-45.
- Virtual Link information – see “Displaying OSPF Virtual Link Information” on page 26-47.
- ABR and ASBR information – see “Displaying OSPF ABR and ASBR Information” on page 26-44.
- Trap state information – see “Displaying OSPF Trap Status” on page 26-45.

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 26.1: 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 26.2: 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

Table 26.2: CLI Display of OSPF Area Information (Continued)

This Field...	Displays...
Cost	The area's cost.
SPFR	The SPFR value.
ABR	The ABR number.
ASBR	The ABSR number.
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
```

Syntax: show ip ospf neighbor [router-id <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 26.3: 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 26.3: 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 Foundry 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 26.4: 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

Table 26.4: Output of the show ip ospf interface command

This field	Displays
Type	The area type, which can be one of the following: • Broadcast = 0x01 • Point to Point = 0x03 • Virtual Link = 0x04
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 26.5: 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 Foundry technical support.
Tag	The external route tag.
Flags	State information for the route entry. This information is used by Foundry 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

Table 26.5: CLI Display of OSPF Route Information (Continued)

This Field...	Displays...
State	State information for the path. This information is used by Foundry 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 **router-id** <ip-addr> parameter shows the External LSAs for the specified OSPF router.

The **sequence-number** <num(Hex)> parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

This display shows the following information.

Table 26.6: 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.
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 Foundry 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-state** 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 26.7: 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 26.8: CLI Display of OSPF Border Routers

This Field...	Displays...
(Index)	Displayed index number of the border router.
Router ID	ID of the OSPF router
Router type	Type of OSPF router: ABR or ASBR
Next hop router	ID of the next hop router
Outgoing interface	ID of the interface on the router for the outgoing route.
Area	ID of the OSPF area to which the OSPF router belongs

Displaying OSPF Trap Status

All traps are enabled by default when you enable OSPF. To disable or re-enable an OSPF trap, see “Modify OSPF Traps Generated” on page 26-30.

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

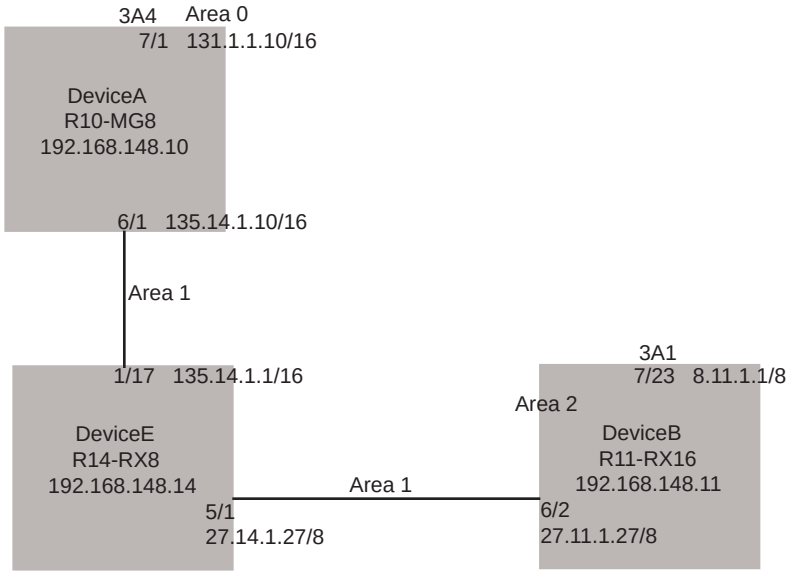
You can display OSPF virtual neighbor and virtual link information. For example, the following show run display shows the configuration in Figure 26.9.

```
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 26.9 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 26.9.

```
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 26.9.

```
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.

Chapter 27
Configuring BGP4 (IPv4)

This chapter provides details on how to configure ***Border Gateway Protocol version 4 (BGP4)*** on the BigIron RX:

BGP commands that are supported in IPv4 are listed in Table 27.1.

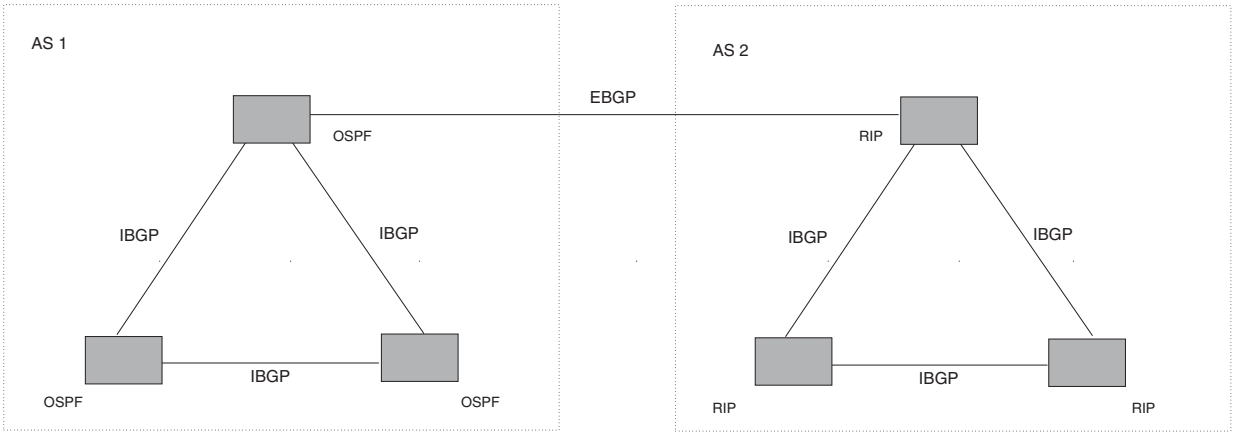
Overview of BGP4

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 27.1 on page 27-2 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 BigIron RX can be configured to redistribute routes among BGP4, ISIS, RIP, and OSPF. They also can redistribute static routes.

Figure 27.1 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 BigIron RX 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 BigIron RX 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 BigIron RX exchanges complete BGP4 route tables with the neighbor. After this initial exchange, the BigIron RX 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. See “BGP4 Message Types” on page 27-5 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 route.

   NOTE: By default, the device does not use the default route to resolve BGP4 next hop. Also see “Enabling Next-Hop Recursion” on page 27-34 and “Using the IP Default Route as a Valid Next Hop for a BGP4 Route” on page 27-33.

2. Prefer the route that was originated locally (by this BGP4 BigIron RX).
3. Use the path with the largest weight.
4. If the weights are the same, prefer the route with the largest local preference.
5. If the local preferences are the same, prefer the route 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 route 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 routes have the same origin type, prefer the route with the lowest MED. For a definition of MED, see “Configuring the BigIron RX To Always Compare Multi-Exit Discriminators (MEDs)” on page 27-15”.
   - 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 BigIron RX 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 route, when comparing the MEDs of the routes.

   NOTE: MED comparison is not performed for internal routes originated within the local AS or confederation.

8. Prefer routes 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.

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 route that comes from the BGP4 router with the lowest router ID, if **compare-router ID** is enabled; otherwise, select the router that is the first in the list.

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 BigIron RX.
- 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 BigIron RX 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 BigIron RX 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 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, see “Changing the Router ID” on page 27-40.
- 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

1. If the internal paths also are the same and BGP4 load sharing is enabled, load share among the paths.

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.

2. Prefer the route that comes from the BGP4 router with the lowest router ID, if **compare-router ID** is enabled; otherwise, select the router that is the first in the list.

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 BigIron RX.
- 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 BigIron RX 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 BigIron RX 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 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, see “Changing the Router ID” on page 21-20.
- 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 neighbor(s) 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.

Foundry Implementation of BGP4

BGP4 is described in RFC 1771 and the latest BGP drafts. The Foundry 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 BigIron RX 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 BigIron RX 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 27.2 BGP configuration levels

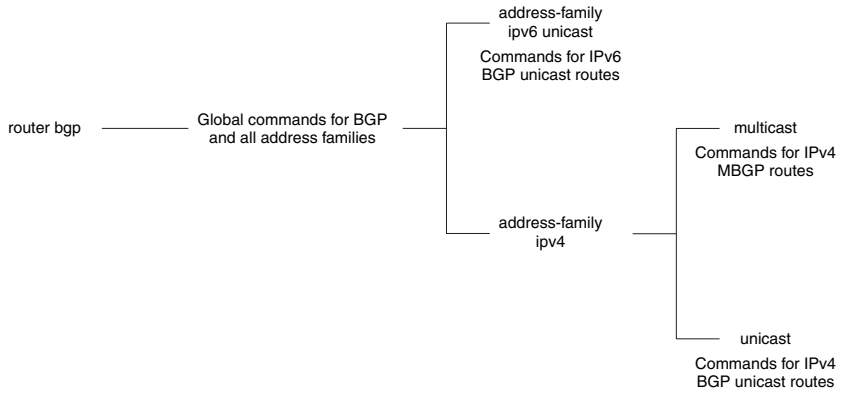


Table 26.1 shows what commands are available at the various BGP configuration levels.

Table 27.1: 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 27-11
address-filter		x		"Filtering Specific IP Addresses" on page 27-12
aggregate-address		x	x	"Aggregating Routes Advertised to BGP4 Neighbors" on page 27-14
always-compare-med	x			"Configuring the BigIron RX To Always Compare Multi-Exit Discriminators (MEDs)" on page 27-15
as-path-filter		x		
as-path-ignore	x			"Disabling or Re-Enabling Comparison of the AS-Path Length" on page 27-15
bgp-redistribute-internal		x		"Redistributing IBGP Routes" on page 27-15
client-to-client-reflection				"Disabling or Re-Enabling Client-to-Client Route Reflection" on page 27-16
cluster-id	x			"Configuring a Route Reflector" on page 27-16
community-filter		x		
compare-routerid	x			"Enabling or Disabling Comparison of the Router IDs" on page 27-16
confederation	x			"Configuring Confederations" on page 27-17
dampening		x	x	"Configuring Route Flap Dampening" on page 27-19
default-information-originate		x	x	"Originating the Default Route" on page 27-20
default-local-preference	x			"Changing the Default Local Preference" on page 27-20
default-metric		x	x	"Changing the Default Metric Used for Redistribution" on page 27-20
distance	x			"Changing Administrative Distances" on page 27-21
enforce-first-as	x			"Requiring the First AS to be the Neighbor's AS" on page 27-22

Table 27.1: IPv4 BGP Commands at Different Configuration Levels (Continued)

Command	Global (iPv4 and IPv6)	IPv4 Address Family Unicast	IPv4 Address Family Multicast	See
exit-address-family	x	x	x	"Entering and Exiting the Address Family Configuration Level" on page 27-11
fast-external-fallover	x			"Enabling Fast External Fallover" on page 27-22
local-as	x			"Setting the Local AS Number" on page 27-22
maximum-paths		x		"Changing the Maximum Number of Shared BGP4 Paths" on page 27-23
med-missing-as-worst	x			"Treating Missing MEDs as the Worst MEDs" on page 27-23
multipath		x		"Customizing BGP4 Load Sharing" on page 27-23
neighbor	x	x	x	"Configuring BGP4 Neighbors" on page 27-24 "Configuring a BGP4 Peer Group" on page 27-30
network		x	x	"Specifying a List of Networks to Advertise" on page 27-32
next-hop-enable-default		x		"Using the IP Default Route as a Valid Next Hop for a BGP4 Route" on page 27-33
next-hop-recursion		x		"Enabling Next-Hop Recursion" on page 27-34
redistribute		x	x	"Modifying Redistribution Parameters" on page 27-36
show	x	x	x	"Displaying BGP4 Information" on page 27-69
table-map		x	x	"Using a Table Map To Set the Tag Value" on page 27-38
timers	x			"Changing the Keep Alive Time and Hold Time" on page 27-39
update-time		x	x	"Changing the BGP4 Next-Hop Update Timer" on page 27-39

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. (See "Closing or Resetting a Neighbor Session" on page 27-66.)

- 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, see “Setting the Local AS Number” on page 27-22.)

NOTE: By default, the Foundry 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, see “Changing the Router ID” on page 27-40. 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 BigIron RX 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 BigIron RX 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 BigIron RX applies the filters in numerical order, beginning with the lowest-numbered filter. When a filter match is true, the BigIron RX 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 BigIron RX advertises the route only within the sub-AS. For information about confederations, see "Configuring Confederations" on page 27-17.

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 BigIron RX advertises the route only within the confederation. For information about confederations, see "Configuring Confederations" on page 27-17.

Aggregating Routes Advertised to BGP4 Neighbors

By default, the BigIron RX advertises individual routes for all the networks. The aggregation feature allows you to configure the BigIron RX to aggregate routes in a range of networks into a single network prefix. For example, without aggregation, the BigIron RX 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 BigIron RX 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. See ""Defining Route Maps" on page 27-49 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 BigIron RX 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 BigIron RX 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.

To configure the router to always compare MEDs, enter the following command:

```
BigIron RX(config-bgp)# always-compare-med
```

Syntax: [no] always-compare-med

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 BigIron RX 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 BigIron RX to redistribute the routes.

To enable the BigIron RX 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 on page 27-4 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 BigIron RX selects the path that came from the neighbor with the lower router ID.
- If BGP4 load sharing is enabled, the BigIron RX 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, see “How BGP4 Selects a Path for a Route” on page 27-3.

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 Foundry 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.

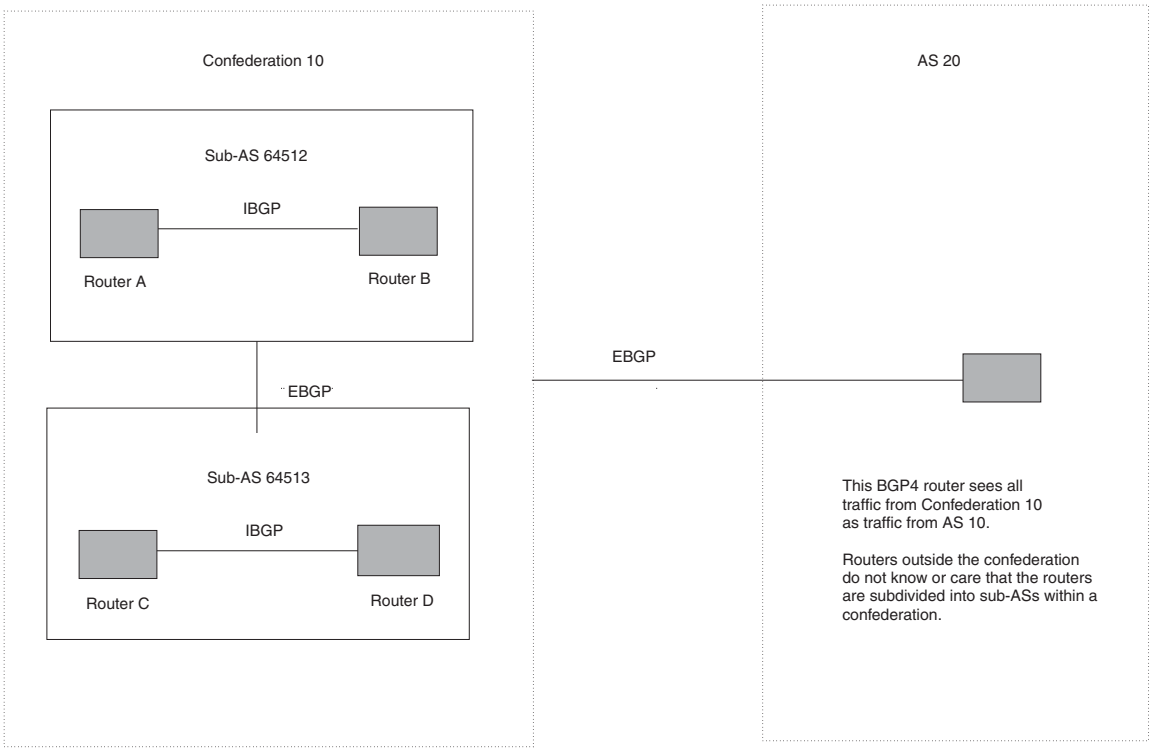
NOTE: Another method for reducing the complexity of an IBGP mesh is to use route reflection. However, if you want to run different Interior Gateway Protocols (IGPs) within an AS, configure a confederation. You can run a separate IGP within each sub-AS.

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, Foundry 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 26.3 shows an example of a BGP4 confederation.

Figure 27.3 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. Foundry 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 BigIron RX 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 BigIron RX 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. See “Defining Route Maps” on page 27-49. See “How BGP4 Selects a Path for a Route” on page 27-3 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 BigIron RX 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 BigIron RX 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 BigIron RX 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. See "How BGP4 Selects a Path for a Route" on page 27-3.

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 BigIron RX:

- 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 BigIron RX 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, see "Changing Administrative Distances" on page 27-21.
- To change the administrative distance for static routes, see "Configuring Static Routes" on page 18-30

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 BigIron RX 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 BigIron RX for this requirement.

When you enable the BigIron RX 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 BigIron RX accepts the Update only if the ASs match. If the ASs do not match, the BigIron RX 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

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 Foundry 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 BigIron RX favors a lower MED over a higher MED during MED comparison. Since the BigIron RX assigns the value 0 to a route path's MED if the MED value is missing, the default MED comparison results in the BigIron RX favoring the route paths that are missing their MEDs.

To change this behavior so that the BigIron RX 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. See "Configuring a BGP4 Peer Group" on page 27-30.

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. See "Administratively Shutting Down a Session with a BGP4 Neighbor" on page 27-32.

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. See "Configuring a BGP4 Peer Group" on page 27-30.

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 BigIron RX sends the IP prefix lists as Outbound Route Filters (ORFs) to the neighbor.
- **receive** – The BigIron RX 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, see "Configuring Cooperative BGP4 Route Filtering" on page 27-55.

NOTE: The current release supports cooperative filtering only for filters configured using IP prefix lists.

default-originate [**route-map** <map-name>] configures the BigIron RX 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. See "Filtering Specific IP Addresses" on page 27-12.

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 BigIron RX 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. See "Filtering AS-Paths" on page 27-43.

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, see "Encryption of BGP4 MD5 Authentication Keys" on page 27-28.

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, see "Defining and Applying IP Prefix Lists" on page 27-48.

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 BigIron RX sends to the neighbor. This option is disabled by default.

route-map in | out <map-name> specifies a route map the BigIron RX 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. See ""Defining Route Maps" on page 27-49.

---

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, see "Configuring a Route Reflector" on page 27-16. 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. See "Using Soft Reconfiguration" on page 27-62.

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, see "Changing the Keep Alive Time and Hold Time" on page 27-39.

unsuppress-map <map-name> removes route suppression from a neighbor's routes when those routes have been suppressed due to aggregation. See "Removing Route Dampening from Suppressed Neighbor's Routes" on page 27-27.

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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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: Foundry 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 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.

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 BigIron RX 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. See “Configuring BGP4 Neighbors” on page 27-24 and “Configuring a BGP4 Peer Group” on page 27-30.

The remaining parameters are the same ones supported for individual neighbors. See “Configuring BGP4 Neighbors” on page 27-24 and “Configuring a BGP4 Peer Group” on page 27-30.

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 BigIron RX 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 BigIron RX, configure the neighbor parameters, then allow the BigIron RX 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 BigIron RX 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 BigIron RX 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, see ““Defining Route Maps” on page 27-49.

Using the IP Default Route as a Valid Next Hop for a BGP4 Route

By default, the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX through an IGP route. This can occur when the IGPs do not learn a complete set of IGP routes, resulting in the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX. 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 BigIron RX 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 BigIron RX recursively looks up the next-hop gateways along the route until the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX can use this route because the BigIron RX 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
```

[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. See "Redistributing OSPF External Routes" on page 27-37.

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. See "Defining Route Maps" on page 27-49 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. See "Defining Route Maps" on page 27-49 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 r**edistribute ospf** only) then only internal OSPF routes will be redistributed.

The **metric** <num> parameter changes the metric. You can specify a value from 0 – 4294967295. The default is not assigned.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the OSPF route to the BGP4 route table.

NOTE: The route map you specify must already be configured on the router. See "Defining Route Maps" on page 27-49 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 BigIron RX 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 BigIron RX 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. See "Defining Route Maps" on page 27-49 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

BigIron RX 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 BigIron RX 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. See “Configuring BGP4 Neighbors” on page 27-24 and “Configuring a BGP4 Peer Group” on page 27-30.

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 BigIron RX 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 BigIron RX. 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 BigIron RX to balance traffic to a route across multiple equal-cost paths of the same type (EBGP or IBGP) for the route.

To configure the BigIron RX 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. See "Changing the Maximum Number of Shared BGP4 Paths" on page 27-23.

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 BigIron RX performs is a comparison of the internal paths.

- When IP load sharing is disabled, the BigIron RX 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 BigIron RX balances the traffic across the multiple paths instead of choosing just one path based on router ID.

See "How BGP4 Selects a Path for a Route" on page 27-3 for a description of the BGP4 algorithm.

When you enable IP load sharing, the BigIron RX 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 BigIron RX, see "Configuring IP Load Sharing" on page 18-38.

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.

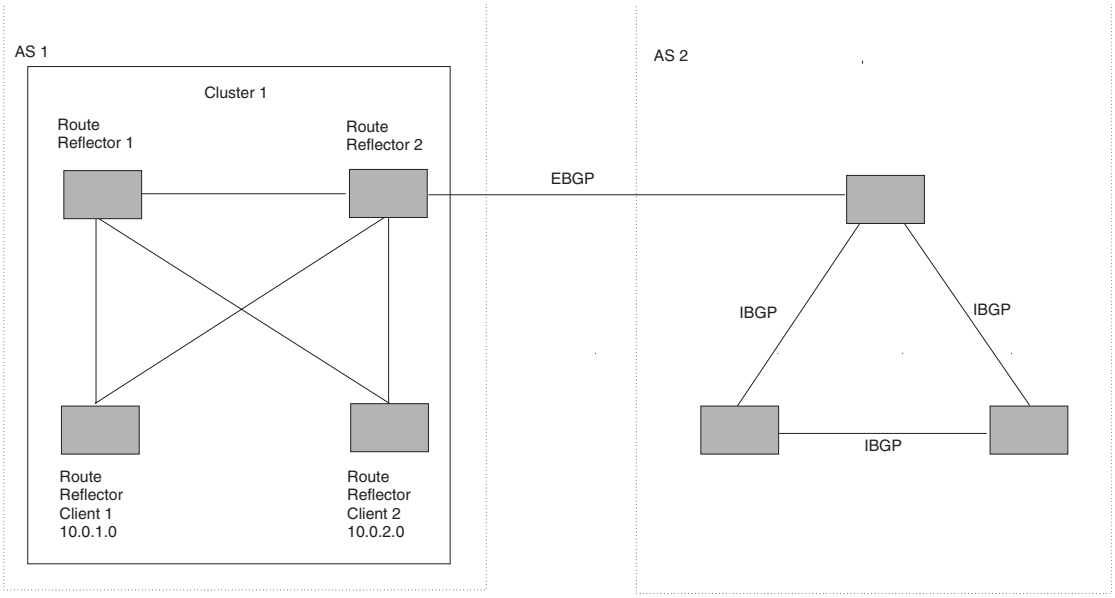
- A ***route reflector*** is an IGP router configured to send BGP route information to all the clients (other BGP4 routers) within the cluster. Route reflection is enabled on all Foundry BGP4 routers by default but does not take effect unless you add route reflector clients to the router.
- A ***route 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 27.4 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX prevents a routing loop.
- The first time a route is reflected by a BigIron RX 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 BigIron RX 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 BigIron RX 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, see “Configuring BGP4 Neighbors” on page 27-24 and “Configuring a BGP4 Peer Group” on page 27-30.

Filtering

This section describes the following:

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 BigIron RX provides the following methods for filtering on AS-path information:

- AS-path filters - see “Defining an AS-Path Filter” on page 27-13.
- 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 BigIron RX 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. See "Matching Based on AS-Path ACL" on page 27-51.

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. See "Configuring BGP4 Neighbors" on page 27-24 and "Configuring a BGP4 Peer Group" on page 27-30.

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 27.2: 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$
_	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_

Table 27.2: BGP4 Special Characters for Regular Expressions (Continued)

Character	Operation
[ ]	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 BigIron RX provides the following methods for filtering on community information:

- Community filters - see “Defining a Community Filter” on page 27-13.
- 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: See “Matching Based on Community ACL” on page 27-52 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. See “Matching Based on Community ACL” on page 27-52

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, see “Using Regular Expressions” on page 27-44. 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 BigIron RX 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 BigIron RX to use IP prefix list Routesfor20 to determine which routes to send to neighbor 10.10.10.1. The BigIron RX 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, see "Configuring BGP4 Neighbors" on page 27-24 and "Configuring a BGP4 Peer Group" on page 27-30.

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 BigIron RX to use ACL 1 to select the routes that the BigIron RX 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 BigIron RX 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. See "Specifying the Match Conditions" on page 27-50 and "Setting Parameters in the Routes" on page 27-53.

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 BigIron RX does not advertise or learn the route.
- If you specify **permit**, the BigIron RX 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. See "Defining an AS-Path ACL" on page 27-44.

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, see "Filtering Specific IP Addresses" on page 27-12.
- To configure an AS-path filter or AS-path ACL, see "Filtering AS-Paths" on page 27-43.
- To configure a community filter or community ACL, see "Filtering Communities" on page 27-46.

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. See "Access Control List" on page 21-1. To configure an IP prefix list, use the **ip prefix-list** command. See "Defining and Applying IP Prefix Lists" on page 27-48.

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 BigIron RX 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. See "Defining an AS-Path ACL" on page 27-44.

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. See “Defining a Community ACL” on page 27-47.

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. See “Access Control List” on page 21-1.

The <name> parameter with the second command specifies an IP prefix list name. To configure an IP prefix list, see “Defining and Applying IP Prefix Lists” on page 27-48.

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. See “Access Control List” on page 21-1.

The <name> parameter with the second command specifies an IP prefix list name. To configure an IP prefix list, see “Defining and Applying IP Prefix Lists” on page 27-48.

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 BigIron RX 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, see "Configuring Route Flap Dampening" on page 27-19.

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. See "Using a Table Map To Set the Tag Value" on page 27-38.

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 BigIron RX performs all filtering of incoming routes locally, on the BigIron RX 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 BigIron RX. Cooperative filtering conserves resources by eliminating unnecessary route updates and filter processing. For example, the BigIron RX can send a deny filter to its neighbor, which the neighbor uses to filter out updates before sending them to the BigIron RX. The neighbor saves the resources it would otherwise use to generate the route updates, and the BigIron RX saves the resources it would use to filter out the routes.

When you enable cooperative filtering, the BigIron RX advertises this capability in its Open message to the neighbor when initiating the neighbor session. The Open message also indicates whether the BigIron RX is configured to send filters, receive filters or both, and the types of filters it can send or receive. The BigIron RX sends the filters as Outbound Route Filters (ORFs) in Route Refresh messages.

To configure cooperative filtering, perform the following tasks on the BigIron RX 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 BigIron RX. You can enable the BigIron RX 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 BigIron RX. Likewise, the BigIron RX 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 BigIron RX to send the IP prefix list as an ORF to neighbor 1.2.3.4. When the BigIron RX sends the IP prefix list to the neighbor, the neighbor filters out the 20.20.0.x routes from its updates to the BigIron RX. (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 BigIron RX sends the IP prefix lists to the neighbor.
- **receive** – The BigIron RX 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 BigIron RX, the BigIron RX 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 BigIron RX 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 BigIron RX.
- The ORFs received from neighbors.

To display the cooperative filtering configuration on the BigIron RX, 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 BigIron RX suppresses unstable routes until the route’s state changes reduce enough to meet an acceptable degree of stability. The Foundry 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 BigIron RX 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 BigIron RX 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 BigIron RX stops using the route. Thus, by default, if a route goes down more than twice, the BigIron RX 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 BigIron RX. If the route's penalty falls below this value, the BigIron RX 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 BigIron RX 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 BigIron RX 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 route(s) 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. See "Using Regular Expressions" on page 27-44.

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**.

This display shows the following information.

Table 27.3: 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). See "Configuring Route Flap Dampening" on page 27-19.

NOTE: The **clear ip bgp damping** command not only clears statistics but also un-suppresses the routes. See "Configuring Route Flap Dampening" on page 27-19.

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. See "Clearing and Resetting BGP4 Routes in the IP Route Table" on page 27-67.

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, see "Configuring BGP4 Neighbors" on page 27-24 and "Configuring a BGP4 Peer Group" on page 27-30.

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 BigIron RX 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, see "Dynamically Refreshing Routes" on page 27-65.

Displaying the Filtered Routes Received from the Neighbor or Peer Group

When you enable soft reconfiguration, the BigIron RX 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 BigIron RX. 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 BigIron RX 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 BigIron RX 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, see "Displaying the BGP4 Route Table" on page 27-89.

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, see "Displaying BGP4 Neighbor Information" on page 27-75.

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 BigIron RX 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 BigIron RX sends a BGP4 OPEN message to a neighbor, the BigIron RX includes a Capability Advertisement to inform the neighbor that the BigIron RX 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 BigIron RX 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 BigIron RX. 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 BigIron RX has stored. Soft reconfiguration does not request additional updates from the neighbor or otherwise affect the session with the neighbor. See “Using Soft Reconfiguration” on page 27-62.
  - 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 BigIron RX’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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX has sent to or received from the neighbor and indicates whether the BigIron RX received confirmation from the neighbor that the neighbor supports dynamic route refresh.

The RefreshCapability field indicates whether this BigIron RX 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 BigIron RX and the neighbor clear all the routes they learned from each other. When the BigIron RX and neighbor establish a new BGP4 session, they exchange route tables again. Use this method if you want the BigIron RX to relearn routes from the neighbor and resend its own route table to the neighbor.
- If you use the soft-outbound option, the BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX. 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 BigIron RX. 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). See "Displaying Route Flap Dampening Statistics" on page 27-98.

NOTE: The **clear ip bgp damping** command not only clears statistics but also un-suppresses the routes. See "Displaying Route Flap Dampening Statistics" on page 27-98.

Removing Route Flap Dampening

You can un-suppress routes by removing route flap dampening from the routes. The BigIron RX 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 route(s) for network 209.157.22.0/24.

Clearing Diagnostic Buffers

The BigIron RX 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 BigIron RX

To display these buffers, use options with the **show ip bgp neighbors** command. See "Displaying BGP4 Neighbor Information" on page 27-75.

This information can be useful if you are working with Foundry 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 BigIron RX. 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
```

This display shows the following information.

Table 27.4: 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. See "Changing the Maximum Number of Shared BGP4 Paths" on page 27-23.
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, see "Displaying the BGP4 Route Table" on page 27-89.
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 router's route-attributes table. To display the route-attribute table, see "Displaying BGP4 Route-Attribute Entries" on page 27-96.
Neighbor Address	The IP addresses of this router's BGP4 neighbors.
AS#	The AS number.

Table 27.4: BGP4 Summary Information (Continued)

This Field...	Displays...

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. See "Administratively Shutting Down a Session with a BGP4 Neighbor" on page 27-32.
	  • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed.
	• ACTIVE – BGP4 is waiting for a TCP connection from the neighbor.
	Note: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.
	• OPEN SENT – BGP4 is waiting for an Open message from the neighbor.
	• OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the router receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle.
	• ESTABLISHED – BGP4 is ready to exchange UPDATE 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 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.

Table 27.4: BGP4 Summary Information (Continued)

This Field...	Displays...
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 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 27.5: 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.

Table 27.5: BGP4 Route Summary Information for a Neighbor (Continued)

This Field...	Displays...
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.
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.

Table 27.5: BGP4 Route Summary Information for a Neighbor (Continued)

This Field...	Displays...
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 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 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 BigIron RX 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. See “Using Soft Reconfiguration” on page 27-62.

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 BigIron RX 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 BigIron RX received better routes from other sources (such as OSPF, RIP, or static IP routes).
- **unreachable** – Displays the routes that are unreachable because the BigIron RX 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 BigIron RX from the neighbor
- Number of routes this BigIron RX 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 27.6: BGP4 Neighbor Information

This Field...	Displays...
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.
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.

Table 27.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
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. See "Administratively Shutting Down a Session with a BGP4 Neighbor" on page 27-32.
	  • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed.
	• ACTIVE – BGP4 is waiting for a TCP connection from the neighbor.
	Note: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.
	• OPEN SENT – BGP4 is waiting for an Open message from the neighbor.
	• OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the router receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle.
	• ESTABLISHED – BGP4 is ready to exchange UPDATE messages with the neighbor.
	  • If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed.
	Note: If you display information for the neighbor using the **show ip bgp neighbor** <ip-addr> command, the TCP receiver queue value will be greater than 0.

Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this router sends keep alive messages to the neighbor. See "Changing the Keep Alive Time and Hold Time" on page 27-39.

Table 27.6: 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. See “Changing the Keep Alive Time and Hold Time” on page 27-39.
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 27.6: 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

Table 27.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...

Last Connection Reset Reason (cont.)	• Reasons specific to the Foundry 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 27.6: 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 27.6: 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.

Table 27.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
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.
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 BigIron RX 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 BigIron RX received better routes from other sources (such as OSPF, RIP, or static IP routes).
- The routes that are unreachable because the BigIron RX does not have a valid RIP, OSPF, or static route to the next hop.
- Routes for a specific network advertised by the BigIron RX 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 BigIron RX 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 27.7: 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.

Table 27.7: 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 Advertised Routes

To display the routes the BigIron RX 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, see Table 26.9 on page 26-95. 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, see Table 26.9 on page 26-95. 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 BigIron RX 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, see Table 26.9 on page 26-95. 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 27.8: 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, see "Using Soft Reconfiguration" on page 27-62.
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.

Table 27.8: BGP4 Summary Route Information (Continued)

This Field...	Displays...
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 27-3 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 BigIron RX 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 BigIron RX 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. See “Using Regular Expressions” on page 27-44.

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 BigIron RX 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, see Table 26.9 on page 26-95. 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, see Table 26.9 on page 26-95. 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 path(s) installed:
         Gateway           Port
         192.168.2.1       2/1
       Route is advertised to 1 peers:
        20.20.20.2(65300)
```

Syntax: show ip bgp [route] <ip-addr>/<prefix> [longer-prefixes] | <ip-addr>

If you use the **route** option, the display for the information is different, as shown in the following example:

```
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 27.9: 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.
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.

Table 27.9: BGP4 Network Information (Continued)

This Field...	Displays...
Status	The route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. **Note**: If the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the 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. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **Note**: If the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **Note**: This field appears only if you enter the **route** option.

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
```


These displays show the following information.

Table 27.10: 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. • B – BEST. BGP4 has determined that this is the optimal route to the destination. **Note**: If the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the 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. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **Note**: If the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
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.

Table 27.10: BGP4 Route Information (Continued)

This Field...	Displays...
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 27.11: 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.

Table 27.11: BGP4 Route-Attribute Entries Information (Continued)

This Field...	Displays...
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • `AS Number` shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • `Router-ID` shows the router that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated *and* this aggregation has resulted in information loss. • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred **Note**: Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.

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. See “Using Regular Expressions” on page 27-44.

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 filter(s) are displayed.

This display shows the following information.

Table 27.12: 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.

Table 27.12: Route Flap Dampening Statistics

This Field...	Displays...
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.

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>]

Chapter 28
Configuring IS-IS (IPv4)

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 Foundry 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 1377 – "The PPP OSI Network Layer Control Protocol (OSINLCP)", 1992.
- 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" (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 Foundry'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 BigIron RX's IP route table that OSPF has to the table. The protocol sends the best IS-IS path to a given destination to the CPU 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

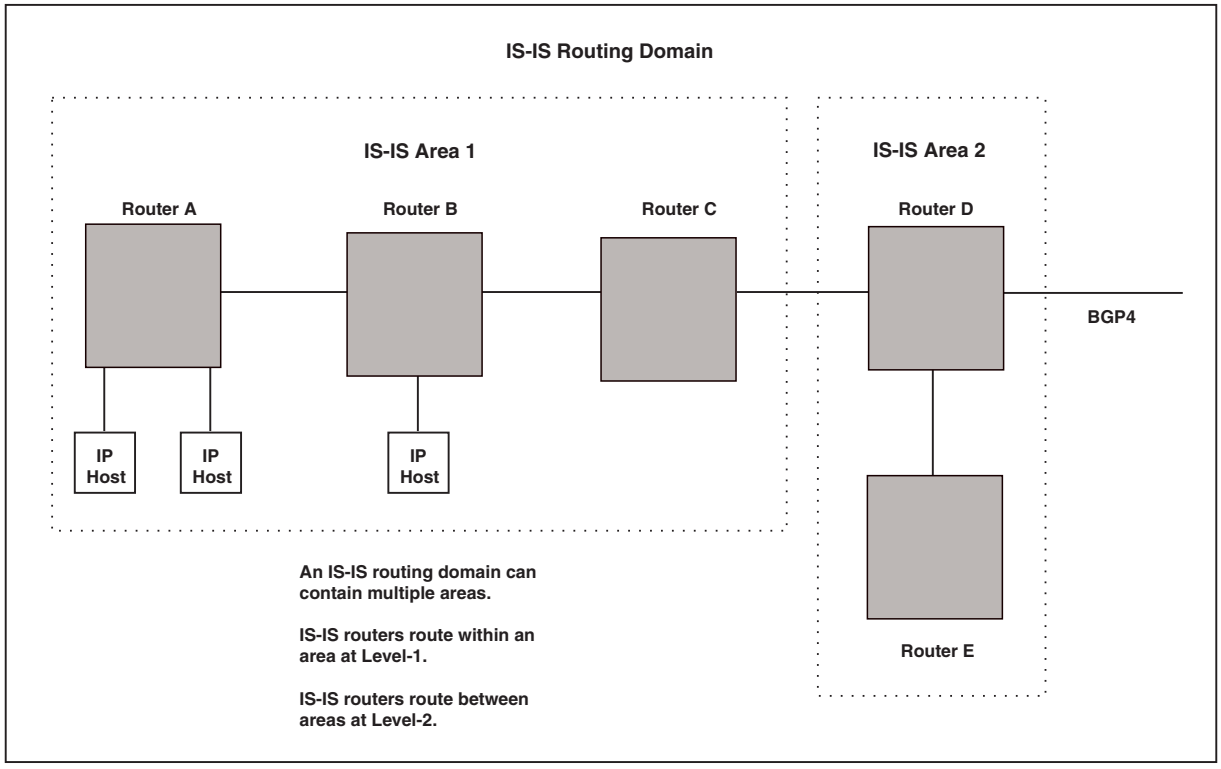
IS-IS uses the following categories to describe devices within an IS-IS routing domain (similar to an OSPF Autonomous System):

- ***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 28.1 shows an example of an IS-IS network.

Figure 28.1 An IS-IS network contains Intermediate Systems (ISs) and host systems



NOTE: Since the Foundry 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 BigIron RX. The NET contains the area ID for the area the NET is in.

In Figure 28.1, 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.

In Figure 28.1 on page 28-2, Routers A and B are Level-1 ISs only. Routers C and D are Level-1 ISs and Level-2 ISs. Router E is a Level-1 ISs 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 BigIron RX forms adjacencies are its ***neighbors***, which are other ISs.

A circuit can be a broadcast circuit or a point-to-point circuit. Foundry IS-IS interfaces are configured by default for broadcast circuits, but you can change the circuit type on an interface to point-to-point. Each end of an IS-IS adjacency must use the same circuit type.

In Figure 28.1 on page 28-2, 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.

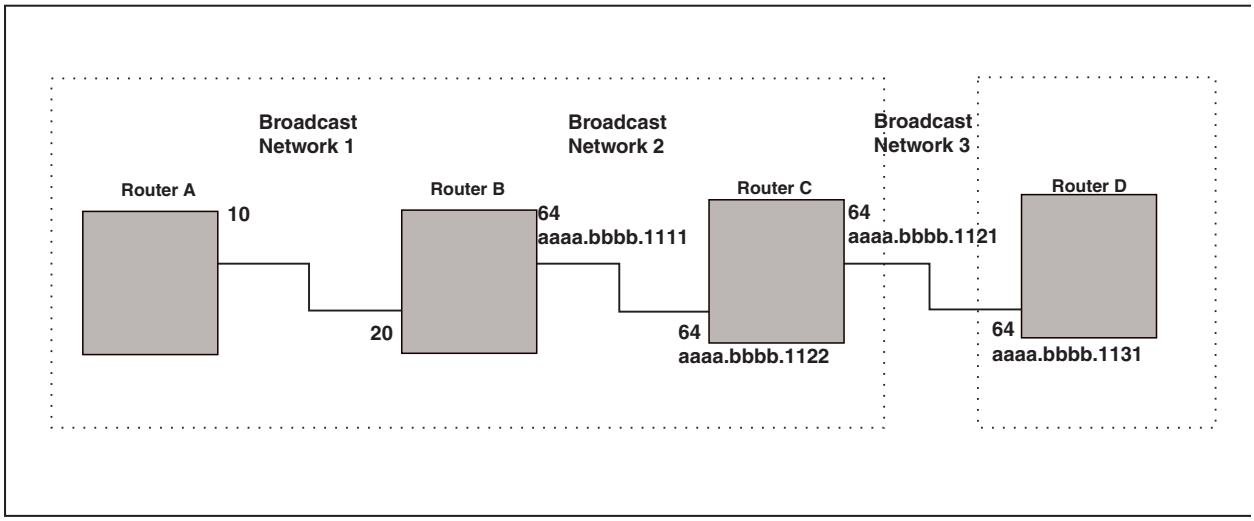
If the Designated IS becomes unavailable (for example, is rebooted), the IS with the next highest priority becomes the new IS. If two or more ISs have the highest priority, the IS with the highest MAC address becomes the Designated IS.

The priority is an interface parameter. Each interface that is enabled for IS-IS can have a different priority.

1.The ISO/IEC specifications use the spelling “routeing”, but this document uses the spelling “routing” to remain consistent with other Foundry documentation.

Figure 28.2 shows an example of the results of Designated IS elections. For simplicity, this example shows four of the five routers in Figure 28.1 on page 28-2, withe same domain and areas.

Figure 28.2 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:

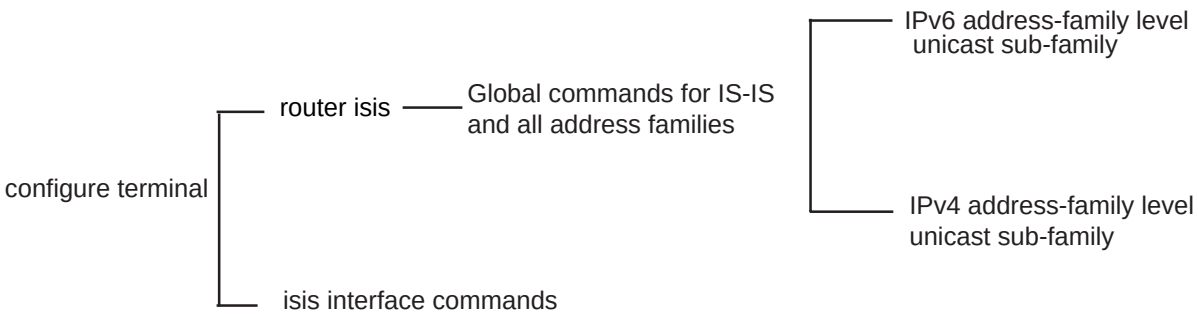
1. Prefer the Level-1 path over the Level-2 path.
2. If there is no Level-1 path, prefer the internal Level-2 path over the external Level-2 path.
3. If there is still more than one path, prefer the path with the lowest metric.
4. If there is more than one path with the lowest metric, load share among 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 28.3 diagrams these levels.

Figure 28.3 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 IPv4. You enter this layer using the **router isis** command.
  - Under the global level, you specify an address family. Address families to separate the IS-IS configurations for IPv4 and IPv6. You enter configurations that are for a specific You enter this level by entering the **address-family** command at the router isis level.
  - Under the address family level, you select a sub-address family, which is the type of routes for the configuration. For IS-IS, you specify **unicast**.
- An interface level.

Global Configuration Level

You enter the global configuration level of ISIS by entering the following command:

```
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 IPv4.

Address Family Configuration Level

Foundry's implementation of IS-IS includes the address family configuration level. Address families allow you to configure IPv4 IS-IS unicast routes that are separate and distinct from IPv6 IS-IS unicast routes, when IPv6 is supported.

Under the address family level, Foundry currently supports the unicast address family configuration level only. The BigIron RX 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 routes.

NOTE: Each address family configuration level allows you to access commands that apply to that particular address family only. To enable a feature in a particular address family, you must specify any associated commands for that feature in that particular address family. You cannot expect the feature, which you may have configured in the IPv4 IS-IS unicast address family, to work in the IPv4 IS-IS unicast address family unless it is explicitly configured in the IPv4 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 change to the interface level for IS-IS configuration, enter the following command.

```
BigIron RX(config)# interface ethernet 2/3
BigIron RX(config-if-e1000-2/3)#ipv4 router isis
```

Syntax: [no] ipv4 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 BigIron RX

   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. See the sections "Globally Configuring IS-IS on a Device" on page 28-7, "Configuring IPv4 Address Family Route Parameters" on page 28-12, and "Configuring ISIS Properties on an Interface" on page 28-17.

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 IPv4.

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 BigIron RX 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 BigIron RX from the network.

In addition, you can configure the BigIron RX 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 BigIron RX to immediately set the overload bit to 1 (on) in all LSPs sent to other ISs.

To configure the BigIron RX 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 BigIron RX to set the overload bit on in all IS-IS LSPs sent to other ISs during the first five seconds following a successful software reload. After the five seconds expire, the BigIron RX stops setting the overload bit on, and instead starts setting the overload bit off.

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 BigIron RX starts performing IS-IS routing immediately following a successful software reload.

Configuring Authentication

By default, the BigIron RX 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 28.1: 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 type of IS-IS packets supported on the device from Level-1 and Level-2 to Level-1 only, enter the following command:

```
BigIron RX(config-isis-router)# is-type level-1-only
```


Syntax: [no] is-type level-1-only | level-1-2 | level-2-only

The **level-1-only | level-1-2 | level-2-only** 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, see "Changing the IS-IS Level on an Interface" on page 28-19.

Disabling or Re-enabling Display of Hostname

Foundry'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 BigIron RX 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 PSNP interval also applies to ISs on a point-to-point network.)

The interval you can configure on the BigIron RX 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 unrefreshed LSP can remain in the Layer 3 Switch'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).

Changing the LSP Refresh Interval

The LSP refresh interval is the maximum number of seconds the BigIron RX 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 General Interval

The LSP general interval is the minimum number of seconds the BigIron RX 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 general 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 seconds 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 <seconds>

Enter 1 – 4294967295 seconds for the LSP interval.

To define an interval for retransmission of LSPs enter a command such as the following:

```
BigIron RX(config-isis-router)#retransmit-interval 3
```

Syntax: [no] retransmit-interval

Enter 0 – 65535 seconds for the retransmission interval.

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 BigIron RX 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 BigIron RX adds extra data to the end of a hello packet to make the packet the same size as the maximum length of PDU the BigIron RX 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 BigIron RX can receive. Other ISs that receive a padded hello PDU from the BigIron RX can therefore ensure that the IS-IS PDUs they send the BigIron RX. Similarly, if the BigIron RX receives a padded hello PDU from a neighbor IS, the BigIron RX knows the maximum size PDU that the BigIron RX can send to the neighbor.

When padding is enabled, the maximum length of a Hello PDU sent by the BigIron RX 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, for example due to point-to-point interoperability issues. If you enable or disable padding on an interface, the interface setting overrides the global setting.

By default, disabling or re-enabling padding affects hello PDUs sent on point-to-point circuits and to an IS-IS broadcast address. You can specify an option to enable or disable the padding for point-to-point or broadcast PDUs.

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 BigIron RX. To re-enable padding, enter the following command:

```
BigIron RX(config-isis-router)# hello padding
```

To disable padding on a specific interface only, enter commands such as the following:

```
BigIron RX(config-isis-router)# interface ethernet 1/1
BigIron RX(config-if-1/1)# hello padding
```

Syntax: [no] hello padding [point-to-point]

The **point-to-point** parameter disables or re-enables the padding only for point-to-point connections.

Enter the **no** form of the command to re-enable hello padding.

To disable hello padding on an interface, see “Disabling and Enabling Hello Padding on an Interface” on page 28-19.

Logging Adjacency Changes

The BigIron RX 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

NOTE: This feature is not supported on Terathon devices.

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-ipv4u)# disable-partial-spf-opt
```

Syntax: [no] disable-partial-spf-opt

This command applies to both IPv4 and IPv4 address families, if both are configured.

Configuring IPv4 Address Family Route Parameters

This section describes how to modify the IS-IS the parameters for the IS-IS IPv4 address family.

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 Layer 3 Switch 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 Layer 3 Switch 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)# metric-style wide
```

This command changes the metric style for both Level-1 and Level-2.

Syntax: [no] metric-style wide [level-1-only | level-2-only]

The **level-1-only | level-1-2 | level-2-only** parameter specifies the level(s) to which the change applies.

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 an amount from 1 – 8. If you change the number of paths to one, the BigIron RX does not load share 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 8
```

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.

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 BigIron RX 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 BigIron RX 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)# 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 BigIron RX 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 RIPng, from OSPFv3, and IPv4 IS-IS to the same destination, and all the paths are using their protocols' default administrative distances, the router selects the OSPFv3 path, because that path has a lower administrative distance than the RIPng and 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
- OSPFv3 – 110
- IPv4 IS-IS – 115
- RIPng – 120
- IBGP – 200
- Local BGP – 200
- Unknown – 255 (the device will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the BigIron RX receives routes for the same network from IPv4 IS-IS and from RIPng, 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 BigIron RX 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> <ip-mask> [level-1-only | level-1-2 | level-2-only]

The <ip-addr> <ip-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-only | level-1-2 | level-2-only** parameter specifies the route types to which the aggregate route applies. The default is **level-2-only**.

Redistributing Routes into 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 BigIron RX can redistribute routes from the following route sources into IPv4 IS-IS:

- BGP4+.
- RIPng.
- OSPFv3.
- Static IPv4 routes.

- IPv4 routes learned from directly connected networks.

The BigIron RX 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 OSPFv3 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, see the "Changing the Default Redistribution Metric" section, which follows this section.

Changing the Default Redistribution Metric

When IPv4 IS-IS redistributes a route from another route source (such as OSPFv3, 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 Foundry 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 1 – 65535. The default is 10.

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 BigIron RX 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 2001:100::/32, enter commands such as the following:


```
BigIron RX(config)# ipv4 access-list static permit any 2001:100::/32
BigIron RX(config)# route-map static permit 1
BigIron RX(config-routemap static)# match ip address static
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
```

For information about the IPv4 ACL and route map syntax, see the “Access Control List” on page 21-1.

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 BigIron RX 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 RIPng Routes into IPv4 IS-IS

To redistribute RIPng 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 BigIron RX to redistribute all RIPng routes into Level-2 IS-IS.

Syntax: [no] redistribute rip [level-1 | level-1-2 | level-2 | metric <number> | metric-type external | internal | route-map <name>]

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing OSPF Version 3 Routes into IPv4 IS-IS

To redistribute OSPFv3 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 OSPFv3 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 BigIron RX 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.

Configuring ISIS Properties on an Interface

This section describe the IS-IS parameters for an interface.

Disabling and ReEnabling 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)# 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.

NOTE: If you have not configured a NET, the software displays the message “ISIS: Please configure NET!” and changes the CLI to the IS-IS configuration level.

Syntax: [no] router isis

Syntax: [no] net <area-id>.<system-id>.<sel>

Syntax: [no] ip router isis

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 Layer 3 Switch’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 1/1. 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 Layer 3 Switch.

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.

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-isis-router)# 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-isis-router)# 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-only | level-2-only]

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-only | level-2-only** 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

Instead of limiting access to an area (level-1) or domain (level-2) you can limit access in forming a connection on a per interface/circuit level by entering a password at the interface level. To enter this password, enter a command such as the following:

```
BigIron RX(config-isis-router)# interface ethernet 2/8
BigIron RX(config-if-e1000-2/8)# isis password
```

Syntax: [no] isis password

Changing the IS-IS Level on an Interface

The section “Changing the IS-IS Level Globally” on page 28-8 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-isis-router)# 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-only | level-1-2 | level-2-only** 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 28-11 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-isis-router)# interface ethernet 2/8
BigIron RX(config-if-e1000-2/8)# hello padding
```

Syntax: [no] hello padding [point-to-point]

The **point-to-point** parameter disables or re-enables the padding only for point-to-point connections.

Enter the **no** form of the command to re-enable 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-isis-router)# 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-only | level-2-only]

The <num> parameter specifies the interval, and can be from 1 – 65535 seconds. The default is 10 seconds.

The **level-1-only | level-2-only** 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-isis-router)# 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-only | level-2-only]

The <num> parameter specifies the multiplier, and can be from 3 – 1000. The default is 3.

The **level-1-only | level-2-only** 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 BigIron RX originates an IS-IS route or calculates a route, the BigIron RX adds a metric (cost) to the route. Each IS-IS interface has a separate metric value. The default is 10.

The BigIron RX 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 BigIron RX 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-isis-router)# interface ethernet 2/8
BigIron RX(config-if-e1000-2/8)# isis metric
```

Syntax: [no] isis metric <num>

The <num> parameter specifies the metric. The range of values you can specify depends on the metric style. You can specify 1 – 63 for the narrow metric style or 1 – 16777215 for the wide metric style. The default in either case is 10.

Displaying IPv4 IS-IS Information

You can display the following information:

- The active configuration (the IS-IS commands in the running-config) – see “Displaying the IS-IS Configuration in the Running-Config” on page 28-21
- Name mappings – “Displaying the Name Mappings” on page 28-21
- Neighbor information – “Displaying Neighbor Information” on page 28-22
- Neighbor adjacency changes – “Displaying IS-IS Syslog Messages” on page 28-23

- Interface information – “Displaying Interface Information” on page 28-25
- Route information – “Displaying Route Information” on page 28-27
- LSP database entries – “Displaying LSP Database Entries” on page 28-28
- Traffic statistics – “Displaying Traffic Statistics” on page 28-32
- Error statistics – “Displaying Error Statistics” on page 28-33

Displaying the IS-IS Configuration in the Running-Config

You can display the global IS-IS configuration commands that are in effect on the Layer 3 Switch 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(config-isis-router)# show isis config

Current IS-IS configuration:
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 an 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(config-isis-router)# show isis hostname
Total number of entries in IS-IS Hostname Table: 1
   System ID       Hostname            * = local IS
  * bbbb.cccc.dddd  BigIron
```

Syntax: show isis hostname

The table in this example contains one mapping, for this Layer 3 Switch. The Layer 3 Switch’s IS-IS system ID is “bbbb.cccc.dddd“ and its hostname is “BigIron”. 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 neighbor**, and **show isis routes** commands uses the host name instead of the system ID. To disable mapping so that these displays use the system ID instead, see “Disabling or Re-enabling Display of Hostname” on page 28-9.

Displaying Neighbor Information

To display IS-IS neighbor information, enter the following command at any level of the CLI:

```
BigIron RX(config-isis-router)# 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

This display shows the following information.

Table 28.2: IS-IS Neighbor Information

This Field...	Displays...
Total number of IS-IS Neighbors	The number of ISs with which the Layer 3 Switch has formed IS-IS adjacencies.
System ID	The System ID of the neighbor.
Interface	The Layer 3 Switch port or virtual interface attached to the neighbor.
SNPA	The Subnetwork Point of Attachment (SNPA), which is the MAC address of the Layer 3 Switch 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 time between transmission of IS-IS hello messages.
Type	The IS-IS type of the adjacency. The type can be one of the following: • ISL1 – Level-1 IS • ISL2 – Level-2 IS • PTP – Point-to-Point IS • ES – ES **Note**: The Layer 3 Switch forms a separate adjacency for each IS-IS type. Thus, if the Layer 3 Switch 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 Layer 3 Switch generates Syslog messages and SNMP traps for the following IS-IS events:

- Overload state (the Layer 3 Switch entering or leaving the overload state)
- Memory overrun (IS-IS is demanding more memory than is available)

You also can enable the Layer 3 Switch to generate Syslog messages and SNMP traps when an adjacency with a neighbor comes up or goes down. To enable logging of adjacency changes, see “Logging Adjacency Changes” on page 28-11.

To display Syslog entries, enter the following command at any level of the CLI:

```
BigIron RX(config-isis-router)# 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 entries):
00d00h00m42s:N:BGP Peer 192.147.202.10 UP (ESTABLISHED)
00d00h00m18s:N:ISIS  L2 ADJACENCY  UP 1234.1234.1234 on circuit 2
00d00h00m08s:N:ISIS  L1 ADJACENCY  UP 1234.1234.1234 on circuit 2
00d00h00m08s:N:ISIS  L2 ADJACENCY  UP 0000.86de.5520 on circuit 1
00d00h00m00s:I:Warm start
```

The messages in this example indicate that the software has been reloaded (Warm start) and adjacencies between the Layer 3 Switch and three ISs have come up.

Syntax: show logging

Table 28.3 lists the IS-IS Syslog messages.

Table 28.3: 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 circuit <circuit-id>	The Layer 3 Switch’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 Layer 3 Switch’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.

Table 28.3: IS-IS Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	ISIS L2 ADJACENCY DOWN <system-id> on circuit <circuit-id>	The Layer 3 Switch'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 Layer 3 Switch'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 Layer 3 Switch has set the overload bit to on (1), indicating that the Layer 3 Switch's IS-IS resources are overloaded.
Notification	ISIS EXITING FROM OVERLOAD STATE	The Layer 3 Switch has set the overload bit to off (0), indicating that the Layer 3 Switch's IS-IS resources are no longer overloaded.

Displaying Interface Information

To display information about the BigIron RX's IS-IS interfaces, enter the following command at any level of the CLI:

```
BigIron RX(config-isis-router)# show isis interface

Total number of IS-IS Interfaces: 2

Interface : 2/4     Local Circuit Number: 00000001
   Circuit Type : BCAST Circuit Mode : LEVEL-1-2
   Circuit State: UP Passive State: FALSE
   MTU : 1497
   Authentication password: abracadabra
   Level-1 Metric: 10, Level-1 Priority: 64
   Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
   Level-1 Designated IS: 00e0.52b5.7800.01-00  Level-1 DIS Changes: 4
   Level-2 Metric: 10, Priority: 64
   Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
   Level-2 Designated IS: 00e0.52b5.7800.01-00, Level-2 DIS Changes: 5
   Next IS-IS LAN Level-1 Hello in 3 seconds
   Next IS-IS LAN Level-2 Hello in 8 seconds
   Number of active level-1 adjacencies: 1
   Number of active level-2 adjacencies: 1
   Circuit State Changes: 0 Circuit Adjacencies State Changes: 2
   Rejected Adjacencies: 0
   Circuit Authentication Fails: 0 Bad LSP: 0
   Control Messages Sent: 204 Control Messages Received: 1990
   IP Address and Subnet Mask:
     128.1.1.2          255.255.255.0
...
```

Syntax: show isis interface

This display shows the following information.

Table 28.4: 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 can be one of the following: • BCAST– broadcast • PTP – point-to-point

Table 28.4: IS-IS Interface Information (Continued)

This Field...	Displays...
Circuit Mode	The IS-IS type in use on the circuit. The mode can be one of the following: • LEVEL-1 • LEVEL-2 • LEVEL-1-2
Circuit State	The state of the circuit, which can be one of the following: • DOWN • UP
Passive State	The state of the passive option, which determines whether the interface is allowed to form an IS-IS adjacency with the IS at the other end of the circuit. The state can be one of the following: • FALSE – The passive option is disabled. The interface can form an adjacency with the IS at the other end of the link. • TRUE – The passive option is enabled. The interface cannot form an adjacency, but can still advertise itself into the area.
MTU	The maximum length supported for IS-IS PDUs sent on this interface.
Authentication Password	The password assigned to the IS-IS interface.
Level-1 Metric	The default-metric value that the Layer 3 Switch inserts in IS-IS Level-1 PDUs originated on this interface.
Level-1 Priority	The priority of this IS to be elected as the Designated IS for Level-1 in this broadcast network.
Level-1 Hello Interval	The number of seconds the software waits between sending Level-1 hello PDUs to the IS at the other end of the circuit.
Level-1 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time for Level-1 Hello messages received on the circuit.
Level-1 Designated IS	The NET of the Level-1 Designated IS.
Level-1 DIS Changes	The number of times the NET of the Level-1 Designated IS has changed.
Level-2 Metric	The default-metric value that the Layer 3 Switch inserts in IS-IS Level-2 PDUs originated on this interface.
Level-2 Priority	The priority of this IS to be elected as the Designated IS for Level-2 in this broadcast network.
Level-2 Hello Interval	The number of seconds the software waits between sending Level-2 Hello messages to the IS at the other end of the circuit.
Level-2 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time for Level-2 LSPs received on the circuit.
Level-2 Designated IS	The NET of the Level-2 Designated IS.

Table 28.4: IS-IS Interface Information (Continued)

This Field...	Displays...
Level-2 DIS Changes	The number of times the NET of the Level-2 Designated IS has changed.
Next IS-IS LAN Level-1 Hello	Number of seconds before next Level-1 Hello message will be transmitted by the Layer 3 Switch.
Next IS-IS LAN Level-2 Hello	Number of seconds before next Level-2 Hello message will be transmitted by the Layer 3 Switch.
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 Layer 3 Switch.
Circuit Authentication Fails	The number of times the Layer 3 Switch rejected a circuit because the authentication did not match the authentication configured on the Layer 3 Switch.
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 Address and Subnet Mask	The IP address and subnet mask configured on this interface.

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 Layer 3 Switch's IS-IS route table, enter the following command at any level of the CLI:

```
BigIron RX(config-isis-router)# show isis routes

Total number of IS-IS routes: 26
Destination      Mask             Cost  Type Tag      Flags1    Flags2
50.50.15.0       255.255.255.0    11    L2   00000000 00000640  73010000
   Path: 1 Next Hop IP: 128.1.1.1         Interface: 2/4   Flags :84000003
50.50.18.0       255.255.255.0    11    L2   00000000 00000640  73010000
   Path: 1 Next Hop IP: 128.1.1.1         Interface: 2/4   Flags :84000003
50.50.21.0       255.255.255.0    11    L2   00000000 00000640  73010000
   Path: 1 Next Hop IP: 128.1.1.1         Interface: 2/4   Flags :84000003
```

Syntax: show isis routes

This display shows the following information.

Table 28.5: IS-IS Route Information

This Field...	Displays...
Total number of IS-IS routes	The total number of routes in the Layer 3 Switch'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.
Flags1	Values used by Foundry technical support for troubleshooting.
Flags2	Values used by Foundry technical support for troubleshooting.
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 from 1 up to 4. When IP load sharing is enabled, the Layer 3 Switch 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 Layer 3 Switch interface (port or virtual interface) attached to the next hop.
Flags	Values used by Foundry 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(config-isis-router)# show isis database

IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
00e0.5200.0001.00-00* 0x00000009   0x027b        1082              0/0/1
00e0.52b5.7800.00-00  0x00000007   0x8631        1014              0/0/0
00e0.52b5.7800.01-00  0x00000006   0xcb17        1014              0/0/0

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
00e0.5200.0001.00-00* 0x0000000a   0xc1da        1082              0/0/1
00e0.52b5.7800.00-00  0x00000005   0xf307        115               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.

The **detail** parameter displays detailed information about the LSPs. See "Displaying Detailed Information" on page 28-30.

The **l1** and **level1** parameters display the Level-1 LSPs only. You can use either parameter. They do the same thing.

The **l2** and **level2** parameters display the Level-2 LSPs only. You can use either parameter. They do the same thing.

The **show isis database** summary display shows the following information.

Table 28.6: 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 Layer 3 Switch to verify that the LSP was not corrupted during transmission over the network.

Table 28.6: IS-IS Summary LSP Database Information (Continued)

This Field...	Displays...
LSP Holdtime	The maximum number of seconds during which the LSP will remain valid. **Note**: The IS that originates the LSP starts the timer for the LSP. As a result, LSPs do not all have the same amount of time remaining when they enter the Layer 3 Switch's LSP database.
ATT	A 4-bit value extracted from bits 4 – 7 in the Attach field of the LSP.
P	The value in the Partition option field of the LSP. The field can have one of the following values: • 0 – The IS that sent the LSP does not support partition repair. • 1 – The IS that sent the LSP supports partition repair.
OL	The value in the LSP database overload field of the LSP. The field can have one of the following values: • 0 – The overload bit is off. • 1 – The overload bit is on, indicating that the IS that sent the LSP is overloaded and should not be used as a Level-2 router.

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(config-isis-router)# show isis database detail

IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
00e0.5200.0001.00-00* 0x00000009   0x027b        1092              0/0/1
   Area Address:  20.8101
   NLPID:  cc
   IP address:  128.1.1.2
   Metric:  10    IP-Extended 128.1.1.0/24  UP bit: 0
   Metric:  10    IS 00e0.52b5.7800.01
   Metric:  10    IS-Extended 00e0.52b5.7800.01

LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
00e0.52b5.7800.00-00  0x00000007   0x8631        1024              0/0/0
   Area Address:  20.8101
   NLPID:  cc
   IP address:  128.1.1.1
   Metric:  10    IP-Internal 2.2.5.0          255.255.255.0
   Metric:  10    IP-Internal 2.2.6.0          255.255.255.0
   Metric:  10    IP-Internal 2.2.7.0          255.255.255.0
   Metric:  10    IP-Internal 2.2.8.0          255.255.255.0
   Metric:  10    IP-Internal 40.1.3.0         255.255.255.0
   Metric:  10    IP-Internal 128.1.1.0        255.255.255.0
   Metric:  10    IS 00e0.52b5.7800.01
   Metric:  10    IS 00e0.52b5.7800.02
```


Syntax: show isis database detail [l1 | l2 | level1 | level2]

The **detail** parameter displays detailed information about the LSPs. If you leave this parameter out, only summary information is displayed.

The **l1** and **level1** parameters display the Level-1 LSPs only. You can use either parameter. They do the same thing.

The **l2** and **level2** parameters display the Level-2 LSPs only. You can use either parameter. They do the same thing.

To display details about Level-1 or Level-2 LSPs only, use a combination of display options, as in the following example:

```
BigIron RX(config-isis-router)# show isis database level2 detail
```

This command displays detailed information for the Level-2 LSPs only.

The **show isis database detail** display shows the following information.

Table 28.7: 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” but can also be “iso”.
IP address	The IP address of the interface that sent the LSP. The Layer 3 Switch can use this address as the next hop in routes to the addresses listed in the rows below.

Table 28.7: IS-IS Detailed LSP Database Information (Continued)

This Field...	Displays...

Destination addresses	The rows of information below the IP address row are the destinations advertised by the LSP. The Layer 3 Switch 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(config-isis-router)# show isis traffic
                            Message Received     Message Sent
 Level-1 Hellos             1029                 115
 Level-2 Hellos             1027                 112
 PTP Hellos                 0                    0
 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 28.8: IS-IS Traffic Statistics

This Field...	Displays...
Level-1 Hellos	The number of Level-1 hello PDUs sent and received by the Layer 3 Switch.
Level-2 Hellos	The number of Level-2 hello PDUs sent and received by the Layer 3 Switch.
PTP Hellos	The number of point-to-point hello PDUs sent and received by the Layer 3 Switch.
Level-1 LSP	The number of Level-1 link-state PDUs sent and received by the Layer 3 Switch.
Level-2 LSP	The number of Level-2 link-state PDUs sent and received by the Layer 3 Switch.
Level-1 CSNP	The number of Level-1 Complete Sequence Number PDUs (CSNPs) sent and received by the Layer 3 Switch.
Level-2 CSNP	The number of Level-2 CSNPs sent and received by the Layer 3 Switch.
Level-1 PSNP	The number of Level-1 Partial Sequence Number PDUs (PSNPs) sent and received by the Layer 3 Switch.
Level-2 PSNP	The number of Level-2 PSNPs sent and received by the Layer 3 Switch.

Displaying Error Statistics

To display IS-IS error statistics, enter the following command at any level of the CLI:

```
BigIron RX(config-isis-router)# 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 28.9: IS-IS Error Statistics

This Field...	Displays...
Area Mismatch	The number of times the Layer 3 Switch interface was unable to create a Level-1 adjacency with a neighbor because the Layer 3 Switch interface and the neighbor did not have any areas in common.
Max Area Mismatch	The number of times the Layer 3 Switch received a PDU whose value for maximum number of area addresses did not match the Layer 3 Switch's value for maximum number of area addresses.
System ID Length Mismatch	The number of times the Layer 3 Switch received a PDU whose ID field was a different length than the ID field length configured on the Layer 3 Switch.
Authentication Fail	The Layer 3 Switch 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 Layer 3 Switch detected a corrupted LSP in the device's memory.
LSP Sequence Number Skipped	The number of times the Layer 3 Switch 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 Layer 3 Switch attempted to set an LSP sequence number to a value higher than the highest number in the CSNP sent by the Designated IS.
Level-1 Database Overload	The number of times the Level-1 state on the Layer 3 Switch changed from Waiting to On or from On to Waiting. • Waiting to On – This change can occur when the Layer 3 Switch 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 Layer 3 Switch's Level-1 LSP database is full and the Layer 3 Switch receives an additional LSP, for which there is no room.
Level-2 Database Overload	The number of times the Level-2 state on the Layer 3 Switch changed from Waiting to On or from On to Waiting. • The change from Waiting to On can occur when the Layer 3 Switch 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 Layer 3 Switch's Level-2 LSP database is full and the Layer 3 Switch receives an additional LSP, for which there is no room.
Our LSP Purged	The number of times the Layer 3 Switch received an LSP that was originated by the Layer 3 Switch itself and had age zero (aged out).

Clearing IS-IS Information

To clear the IS-IS information that the Layer 3 Switch 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 Layer 3 Switch's adjacencies with its IS-IS neighbors)
- Routes
- PDU statistics
- Error statistics

Syntax: clear isis all | counts | neighbor | route | 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 Layer 3 Switch's adjacencies with its IS-IS neighbors and clears the neighbor statistics.

The **route** parameter clears the IS-IS route table.

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 29
Configuring Secure Shell

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 BigIron RX. Foundry'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 BigIron RX negotiates the version of SSHv2 to be used. The highest version of SSHv2 supported by both the BigIron RX 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. See "Using Secure Copy" on page 29-8 for more information.

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 Encryption Algorithms for SSHv2

3DES is the encryption algorithms supported in Foundry's implementation of SSHv2.

Supported MAC (Message Authentication Code) Algorithms

SHA 1 is the MAC algorithm supported in Foundry's implementation of SSHv2:

Configuring SSH

Foundry'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 BigIron RX. The SSH server on the BigIron RX 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. See "Providing the Public Key to Clients" on page 29-3 for an example of what to place in the known hosts file.

While the SSH listener exists at all times, sessions can't be started from clients until a key is generated. Once a key is generated, clients can start sessions. The keys are also not displayed in the configuration file by default. To display the keys, use the **ssh show-host-keys** command in Privileged EXEC mode. To generate a public and private DSA host key pair on a 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 BigIron RX 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 BigIron RX'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 BigIron RX. 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 BigIron RX compares the client's public key to those stored in memory.
3. If there is a match, the BigIron RX uses the public key to encrypt a random sequence of bytes.
4. The BigIron RX 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 BigIron RX 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 BigIron RX.
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 BigIron RX and place all of these keys into one file. This public key file is imported into the BigIron RX.

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.

To cause a public key file called pkeys.txt to be loaded from a TFTP server each time the BigIron RX 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 Foundry device.

The <filename> variable is the name of the dsa public key file that you want to import into the Foundry 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 BigIron RX 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 BigIron RX negotiates a session key and encryption method with the connecting client, user authentication takes place. Foundry'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 BigIron RX. 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; see "Enabling Empty Password Logins" on page 29-5). 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. See "Setting Up Local User Accounts" on page 3-13 for information on setting up user names and passwords on the BigIron RX.

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, Foundry 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 BigIron RX.

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 BigIron RX 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 BigIron RX 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 BigIron RX 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>

See the section “Access Control List” on page 21-1 for details on how to configure ACLs.

Displaying SSH Connection Information

Up to five SSH connections can be active on the BigIron RX. 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    3des-cbc    Mikaila
3          SSH-2    3des-cbc    Jenny
4          SSH-2    3des-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 29.1: 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.
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 BigIron RX, 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 30
Configuring Multi-Device Port Authentication

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 BigIron RX software release 02.2.01.

This chapter is divided into the following sections:

- "How Multi-Device Port Authentication Works" below explains basic concepts about multi-device port authentication.
- "Configuring Multi-Device Port Authentication" on page 30-3 describes how to set up multi-device port authentication on BigIron RX using the Command Line Interface (CLI).
- "Displaying Multi-Device Port Authentication Information" on page 30-8 describes the commands used to display information about a multi-device port authentication configuration.

How Multi-Device Port Authentication Works

The multi-device port authentication feature is a mechanism by which incoming traffic originating from a specific MAC address is switched or forwarded by the device only if the source MAC address is successfully authenticated by a RADIUS server. The MAC address itself is used as the username and password for RADIUS authentication; the user does not need to provide a specific username and password to gain access to the network. If RADIUS authentication for the MAC address is successful, traffic from the MAC address is forwarded in hardware.

If the RADIUS server cannot validate the user's MAC address, then it is considered an authentication failure, and a specified authentication-failure action can be taken. The default authentication-failure action is to drop traffic from the non-authenticated MAC address in hardware. You can also configure the device to move the port on which the non-authenticated MAC address was learned into a restricted or "guest" VLAN, which may have limited access to the network.

RADIUS Authentication

The multi-device port authentication feature communicates with the RADIUS server to authenticate a newly found MAC address. The BigIron RX 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 BigIron RXBigIron RX 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 BigIron RX 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 BigIron RX 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 Switches support multi-device port authentication on untagged ports only.

Supported RADIUS Attributes

The BigIron RX 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 BigIron RX a RADIUS Access-Accept message that allows the BigIron RX 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 Switches 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 limited only by the amount of system resources available on the BigIron RX device.

Support for Multi-Device Port Authentication and 802.1X on the Same Interface

On the BigIron RX, 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 BigIron RX consists of the following tasks:

- Enabling multi-device port authentication globally and on individual interfaces
- 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)
- Enabling denial of service attack protection (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.

Specifying the Format of the MAC Addresses Sent to the RADIUS Server

When multi-device port authentication is configured, the BigIron RX 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. If the port is a tagged or dual-mode port, you cannot use a restricted VLAN as the authentication-failure action.

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 BigIron RX a RADIUS Access-Accept message that allows the BigIron RX 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 and can be disabled). See “Dynamic VLAN and ACL Assignments” on page 30-2 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 BigIron RX 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.
- For tagged ports, if the VLAN ID provided by the RADIUS server is valid, then the port is added to the RADIUS-specified VLAN as a tagged 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.
- For tagged or dual-mode ports, if the VLAN ID provided by the RADIUS server does not match the VLAN ID in the tagged packet that contains the authenticated MAC address as its source address, 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 BigIron RX 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 30.1: 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.

Table 30.1: Output from the show auth-mac-address command (Continued)

This Field...	Displays...
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 30.2: 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.
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.

Table 30.2: Output from the show auth-mac-address configuration command (Continued)

This Field...	Displays...
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.
Protectn 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 30.3: 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.
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 30.4: 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 31
Using the MAC Port Security Feature

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 BigIron RX 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 on all interfaces at once:

```
BigIron RX(config)# port security
BigIron RX(config-port-security)# enable
```

To disable the feature on all interfaces at once:

```
BigIron RX(config)# 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: 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 on all interfaces:

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

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 BigIron RX 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

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 Autosaved MAC Addresses

To display the secure MAC addresses that have been saved to the configuration by the autosave feature, enter the following command:

```
BigIron RX# show port security autosave
```

Syntax: show port security autosave

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 31.1: 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 31.2: 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 31.3: 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 31.4: 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 32
Configuring 802.1X Port Security

The BigIron RX software release 02.2.01 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 doesn't 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.

This chapter contains the following sections:

- "How 802.1X Port Security Works" on page 32-1 explains basic concepts about 802.1X port security.
- "Configuring 802.1X Port Security" on page 32-7 describes how to set up 802.1X port security on BigIron RX devices using the Command Line Interface (CLI).
- "Displaying 802.1X Information" on page 32-18 describes the commands used to display information about an 802.1X port security configuration.
- "Sample 802.1X Configurations" on page 32-25 shows diagrams of two 802.1X port security configurations and the CLI commands used for implementing them.

IETF RFC Support

Foundry'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

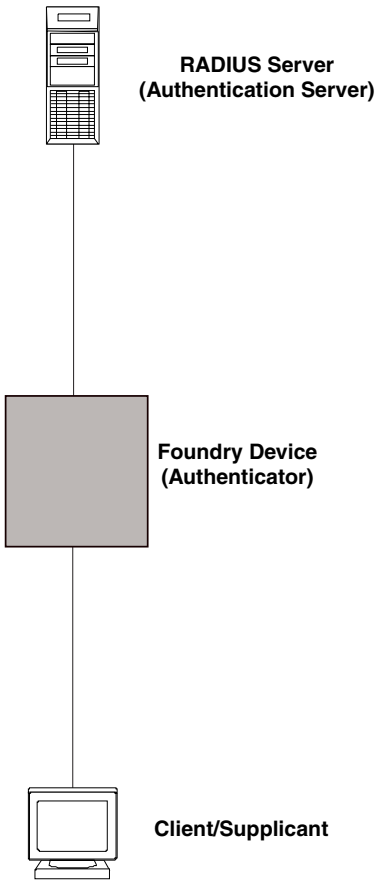
The 802.1X standard defines the roles of ***Client/Supplicant***, ***Authenticator***, and ***Authentication Server*** in a network.

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 32.1 illustrates these roles.

Figure 32.1 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.

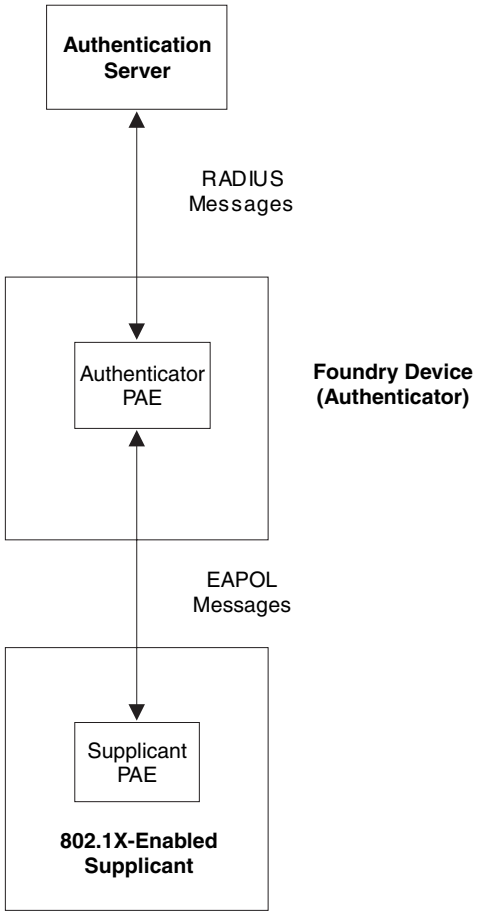
Authentication Server – The device that validates the Client and specifies whether or not the Client may access services on the device. Foundry supports Authentication Servers running RADIUS.

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 32.2 shows the relationship between the Authenticator PAE and the Supplicant PAE.

Figure 32.2 Authenticator PAE and Supplicant PAE



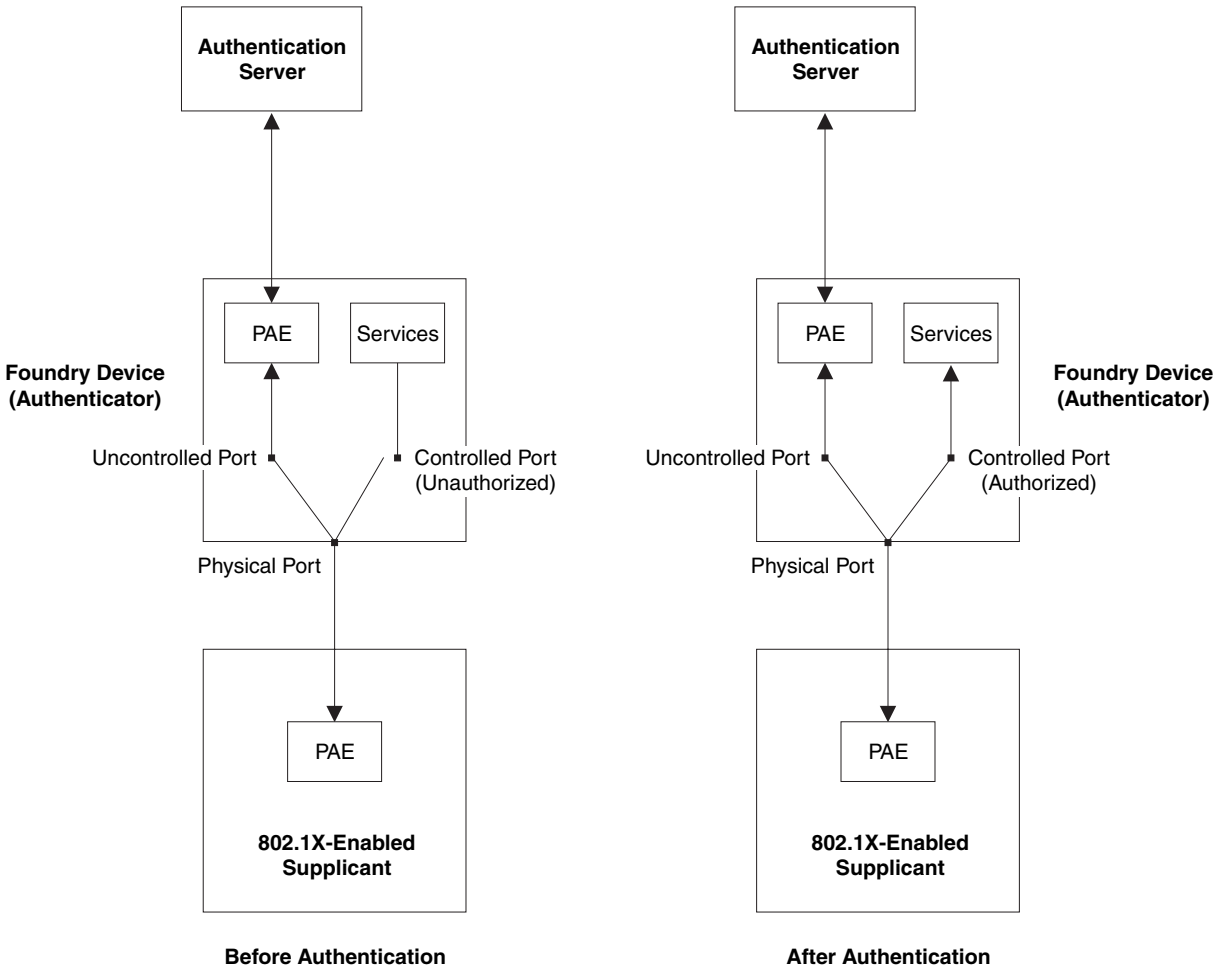
Authenticator PAE – The Authenticator PAE communicates with the Supplicant PAE, receiving identifying information from the Supplicant. Acting as a RADIUS client, the Authenticator PAE passes the Supplicant's information to the Authentication Server, which decides whether the Supplicant can gain access 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 32.3 illustrates this concept.

Figure 32.3 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.

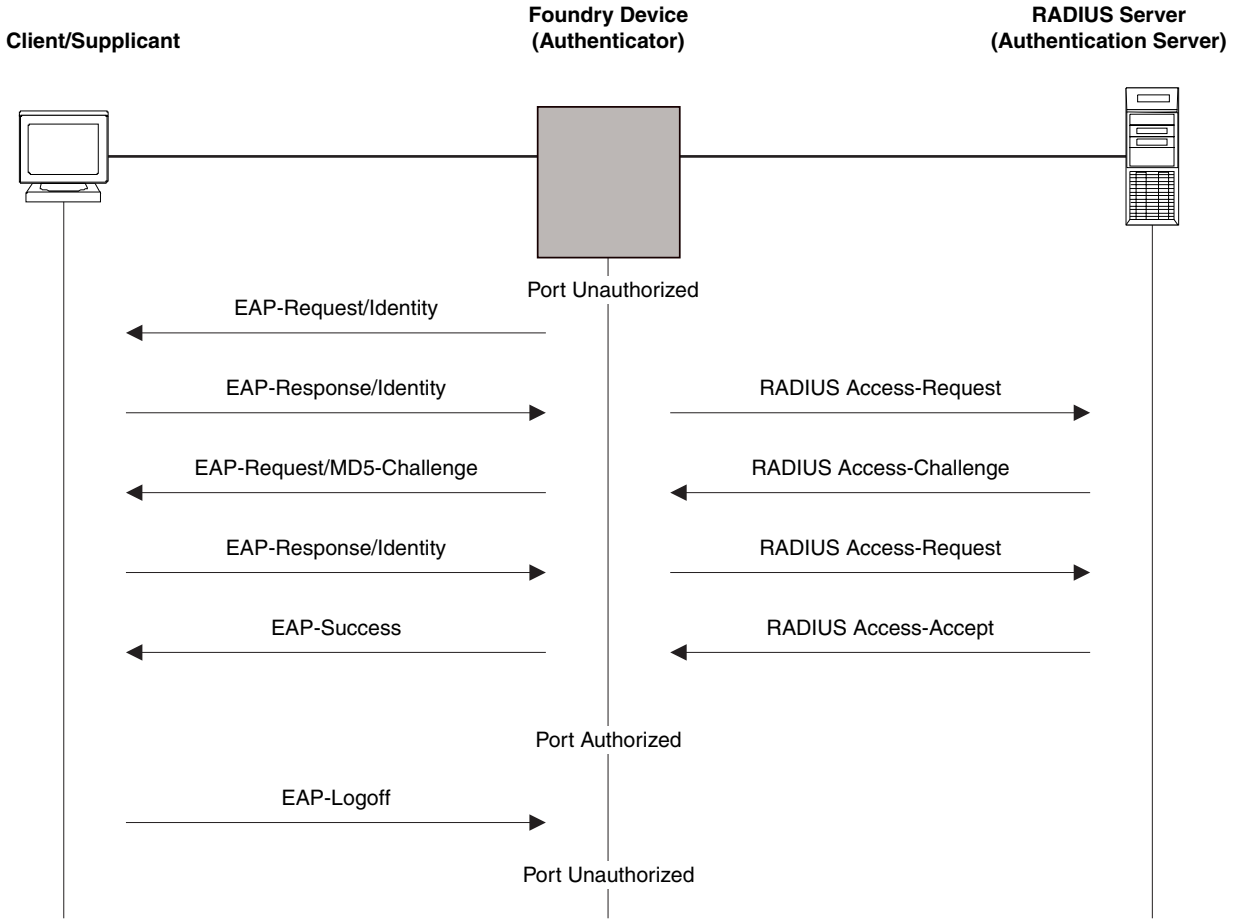
During authentication, EAPOL messages are exchanged between the Supplicant PAE and the Authenticator PAE, and RADIUS messages are exchanged between the Authenticator PAE and the Authentication Server. See "Message Exchange During Authentication" on page 32-4 for an example of this process. If the Client 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 BigIron RX 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. See "Enabling 802.1X Port Security" on page 32-13 for more information.

Message Exchange During Authentication

Figure 32.4 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 32.4 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.

Foundry'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. See "Configuring Dynamic VLAN Assignment for 802.1X Ports" on page 32-9 for more information.

Foundry'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 BigIron RX sends EAP-Request/Identity frames to the Client, but the Client does not respond to them.

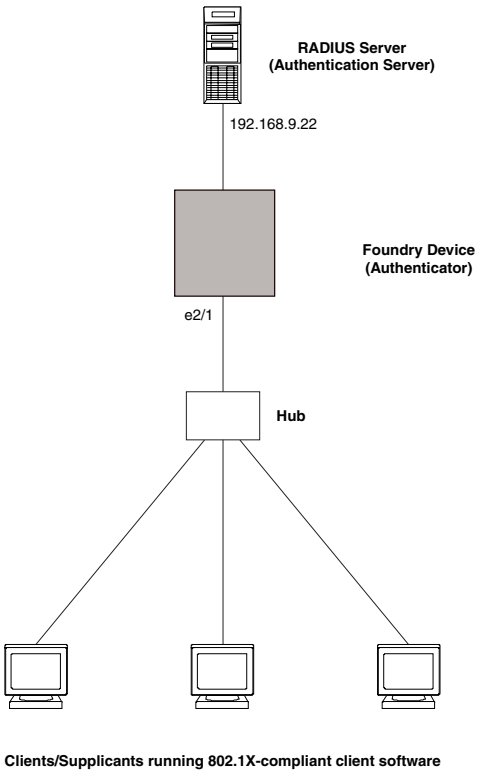
When a Client that supports 802.1X attempts to gain access through a non-802.1X-enabled port, it sends an EAP start frame to the BigIron RX device. When the device does not respond, the Client considers the port to be authorized, and starts sending normal traffic.

BigIron RX devices support Identity and MD5-challenge request types in EAP Request/Response messages.

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 32.5 illustrates a sample configuration where multiple Clients are connected to a single 802.1X port.

Figure 32.5 Multiple Clients connected to a single 802.1X-enabled port



If there are multiple Clients connected to a single 802.1X-enabled port, the BigIron RX 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 BigIron RX 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 32.5), 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 BigIron RX 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 BigIron RX performs 802.1X authentication for the Client. Messages are exchanged between the BigIron RX 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.
    - See "Specifying the Number of Authentication Attempts the Device Makes Before Dropping Packets" on page 32-17 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 BigIron RX 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. See "Clearing a dot1x-mac-session for a MAC Address" on page 32-18 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 and/or outbound port, if that information is available.

For more information on sFlow, see "sFlow" on page 37-1 in the "Remote Network Monitoring" chapter.

Configuring 802.1X Port Security

Configuring 802.1X port security on a BigIron RX 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 32-8
    - "Setting RADIUS Parameters" on page 32-8
    - "Configuring Dynamic VLAN Assignment for 802.1X Ports" on page 32-9 (optional)
2. Configuring the BigIron RX's role as the Authenticator:
    - "Enabling 802.1X Port Security" on page 32-13

- "Initializing 802.1X on a Port" on page 32-17 (optional)

3. Configuring the BigIron RX device's interaction with Clients:
   - "Configuring Periodic Re-Authentication" on page 32-15 (optional)
   - "Re-Authenticating a Port Manually" on page 32-15 (optional)
   - "Setting the Quiet Period" on page 32-15 (optional)
   - "Setting the Interval for Retransmission of EAP-Request/Identity Frames" on page 32-16 (optional)
   - "Specifying the Number of EAP-Request/Identity Frame Retransmissions" on page 32-16 (optional)
   - "Specifying a Timeout for Retransmission of EAP-Request Frames to the Client" on page 32-16 (optional)
   - "Allowing Multiple 802.1X Clients to Authenticate" on page 32-17 (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. See "Support for Multi-Device Port Authentication and 802.1X on the Same Interface" on page 30-3.

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. Foundry 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 BigIron RX. 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

Foundry'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 BigIron RX, 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 BigIron RX, you must add the following attributes to a user's profile on the RADIUS server:

Table 32.1: 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 BigIron RX 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 BigIron RX 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 BigIron RX 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). See "Displaying Dynamically Assigned VLAN Information" on page 32-22 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 Foundry 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 Foundry BigIron RX, then it is considered an authentication failure.
- If the port is a tagged or dual-mode port, and the RADIUS Access-Accept message specifies the name or ID of a valid VLAN on the Foundry BigIron RX, then the port is placed in that VLAN. If the port is already a member of the RADIUS-specified VLAN, no further action is taken. Note that the Client's dot1x-mac-session is set to "access-is-allowed" for the RADIUS-specified VLAN only. If traffic from the Client's MAC address is received on any other VLAN, it is dropped.
- 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.

Using Dynamic VLAN Assignment with the MAC Port Security Feature

The MAC port security feature allows the Foundry BigIron RX to learn a limited number of "secure" MAC addresses on an interface; however, it cannot be enabled on an 802.1X enabled port.

The interface will forward only packets with source MAC addresses that match these secure addresses. If the interface receives a packet with a source MAC address that is different from any of the secure addresses, it is considered a security violation, and subsequent packets from the violating MAC address can be dropped, or the port can be disabled entirely.

If a port has been disabled due to a MAC port security violation, 802.1X clients attempting to connect over the port cannot be authorized. In addition, 802.1X clients connecting from non-secure MAC addresses cannot be authorized.

MAC port security cannot be enabled on dot1x-enabled ports and vice-versa.

To use 802.1X dynamic VLAN assignment with the MAC port security feature on an interface, you must set the number of secure MAC addresses to two or more. For example:

```
BigIron RX(config)# int e 3/2
BigIron RX(config-if-e100-3/2)# port security
BigIron RX(config-port-security-e100-3/2)# maximum 2
BigIron RX(config-port-security-e100-3/2)# exit
```

NOTE: There is small chance that an interface can be inadvertently disabled when both 802.1X (with dynamic VLAN assignment) and the MAC port security feature are enabled on the interface. When this happens, disable then re-enable the interface to bring the interface back up.

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 port 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 BigIron RX 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 Foundry IP ACL or MAC address filter.

The following is the syntax for configuring the Filter-ID attribute to refer to a Foundry 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.2.in	mac filter 2 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800
mac.2.in	mac filter 2 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800
mac.3.in	mac filter 3 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).

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 Foundry 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 Foundry Vendor-Specific attribute on a RADIUS server. These IP ACLs and MAC address filters follow the same syntax as other Foundry ACLs and MAC address filters. See "Access Control List" on page 21-1 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.

See Figure 32.3 on page 32-4 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. 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**, the 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.

NOTE: 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

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 BigIron RX waits a specified amount of time before trying again. The amount of time the BigIron RX waits is specified with the **quiet-period** parameter. 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 MG8, 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 BigIron RX receives RADIUS messages from the RADIUS server, encapsulates them as EAPOL frames, and sends them to the Client. When the BigIron RX 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, 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 32-6 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.

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 32.2: 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”.

Table 32.2: Output from the show dot1x command (Continued)

This Field...	Displays...
system-auth-control	Whether system authentication control is enabled on the device. The **dot1x-enable** command enables system authentication control on the device.
Number of ports enabled	Number of interfaces on the devices that have been enabled for 802.1X.
re-authentication	Whether periodic re-authentication is enabled on the device. See "Configuring Periodic Re-Authentication" on page 32-15. 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 strick 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). See "Setting the Quiet Period" on page 32-15 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). See "Setting the Interval for Retransmission of EAP-Request/Identity Frames" on page 32-16 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. See "Specifying a Timeout for Retransmission of EAP-Request Frames to the Client" on page 32-16 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. See "Specifying a Timeout for Retransmission of Messages to the Authentication Server" on page 32-16 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). See "Specifying the Number of EAP-Request/Identity Frame Retransmissions" on page 32-16 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). See "Configuring Periodic Re-Authentication" on page 32-15 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.

Table 32.2: Output from the show dot1x command (Continued)

This Field...	Displays...
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 32.3: 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. See “Allowing Multiple 802.1X Clients to Authenticate” on page 32-17 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 32.4: 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.

Table 32.4: Output from the show dot1x statistics command (Continued)

This Field...	Displays...
Num sessions	Total number of dot1x sessions, which include authenticated, restricted, denied and sessions in the init 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 <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 Filter" 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 32.5 describes the information displayed by the **show dot1x mac-session** command.

Table 32.5: 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 f	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 32.6: 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.

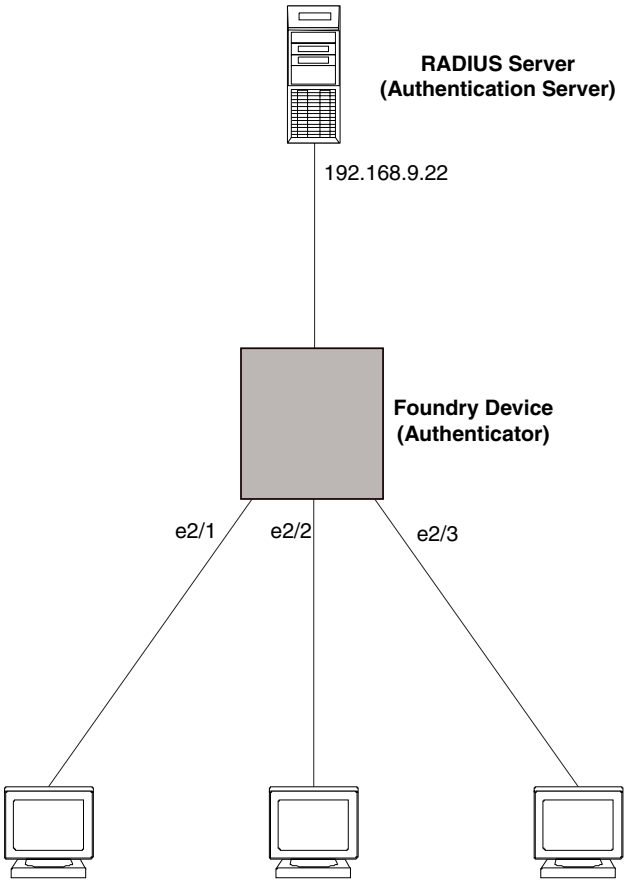
Sample 802.1X Configurations

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 32.6 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 32.6 Sample point-to-point 802.1X configuration



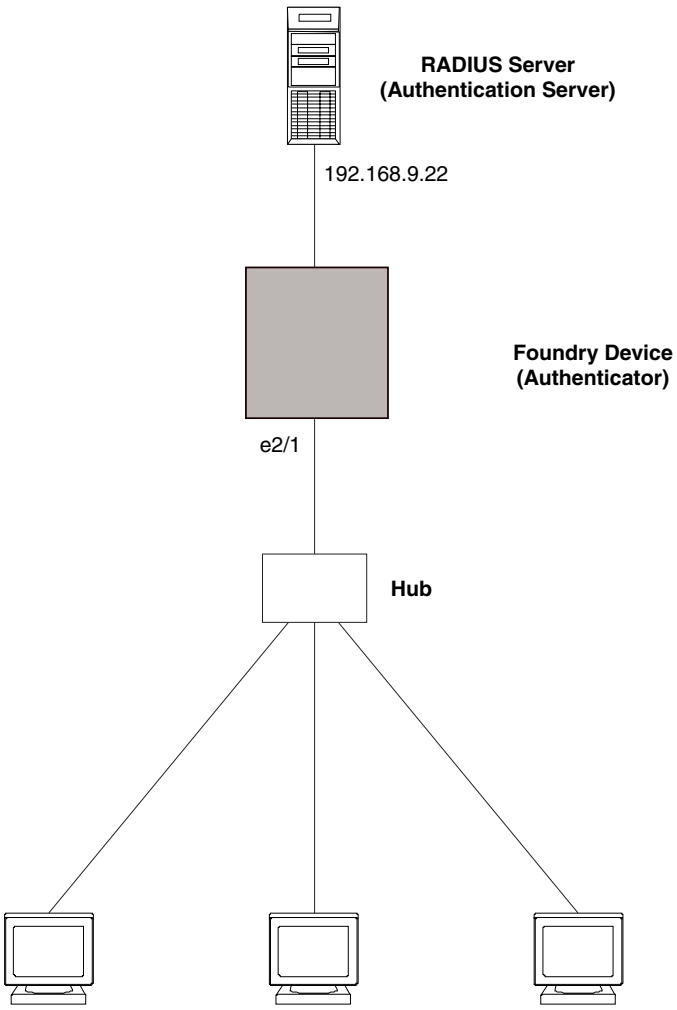
The following commands configure the BigIron RX in Figure 32.6:

```
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 32.7 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 32.6, 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 32.7 Sample 802.1X configuration using a hub



The following commands configure the BigIron RX in Figure 32.7:

```
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)# dot1x multiple-hosts
BigIron RX(config-if-e100-1)# exit
```

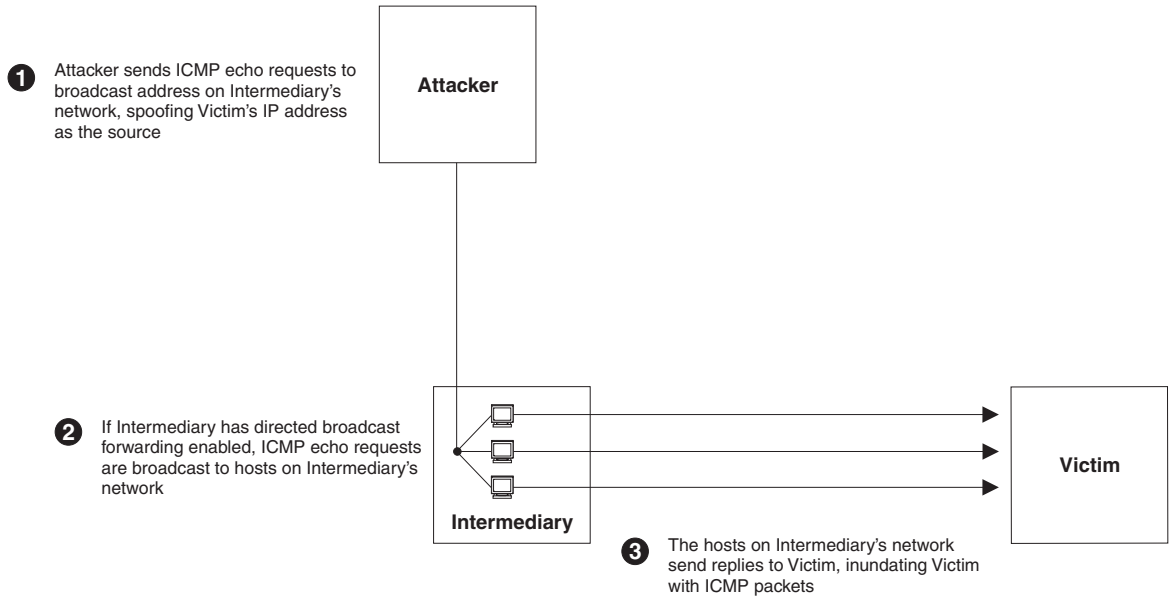
Chapter 33
Protecting Against Denial of Service Attacks

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 33.1 illustrates how a Smurf attack works.

Figure 33.1 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 and/or locked 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 - 00000, 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 physical Ethernet 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. See "Disabling the TCP Security Enhancement" on page 33-4.

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, see "Disabling the TCP Security Enhancement" on page 33-4.

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 BigIron RX 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 BigIron RX 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 BigIron RX sends an acknowledgement (ACK) segment to the peer.

The TCP security enhancement is enabled by default. To disable it, see "Disabling the TCP Security Enhancement" on page 33-4.

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, see "Disabling the TCP Security Enhancement" on page 33-4.

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 34
Securing SNMP Access

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.

The chapter "Securing Access to Management Functions" on page 3-1 introduced a few methods used to secure SNMP access. They included the following:

- "Using ACLs to Restrict SNMP Access" on page 3-6
- "Restricting SNMP Access to a Specific IP Address" on page 3-7
- "Restricting SNMP Access to a Specific VLAN" on page 3-8
- "Disabling SNMP Access" on page 3-10

This chapter presents additional methods for securing SNMP access to the BigIron RX. It contains the following sections:

- "Establishing SNMP Community Strings" on page 34-1
- "Using the User-Based Security Model" on page 34-3
- "Defining SNMP Views" on page 34-8

Restricting SNMP access using ACL, VLAN, or a specific IP address constitute the first level of defense when the packet arrives at a BigIron RX. The next level uses one of the following methods:

- Community string match In SNMP versions 1 and 2
- User-based model in SNMP version 3

SNMP views are incorporated in community strings and the user-based model.

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, see the section “Defining SNMP Views” on page 34-8.

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. See “Using ACLs to Restrict SNMP Access” on page 3-6 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. (See the section “Defining SNMP Views” on page 34-8.)

NOTE: SNMP version 3 Notification is not supported at this time. The system will generate traps in SNMP version 1 format.

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. See “Defining the Engine ID” on page 34-4.
2. Create views that will be assigned to SNMP user groups using the **snmp-server view** command. See the “Defining SNMP Views” on page 34-8 for details.
3. Create ACL groups that will be assigned to SNMP user groups using the **access-list** command. Refer to “Access Control List” on page 21-1 for details.
4. Create user groups using the **snmp-server group** command. See “Defining an SNMP Group” on page 34-4.

5. Create user accounts and associate these accounts to user groups using the **snmp-server user** command. See “Defining an SNMP User Account” on page 34-5.

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
```

See the section “Displaying the Engine ID” on page 34-6 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 Foundry Networks 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. (See “Establishing SNMP Community Strings” on page 34-1.) 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. See the section “Defining SNMP Views” on page 34-8, 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.

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>]]]

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.

Currently, DES is the only encryption type supported for priv password.

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.

Varbind Object Identifier	Description
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 Foundry 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-s group admingrp v3 priv read all write all notify all
BigIron RX(config)#snmp-s user adminuser admingrp v3 auth md5 <auth password> priv
<privacy password>
BigIron RX(config)#snmp-s host <dest-ip> version v3 privacy adminuser
```

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
d32e66152f89de9b2e0cb17a65595f43
```


Chapter 35
Enabling the Foundry Discovery Protocol (FDP) and Reading Cisco Discovery Protocol (CDP) Packets

This chapter discusses the following features:

- Foundry Discovery Protocol (FDP) – a protocol used by Foundry devices to advertise themselves to other Foundry devices
- Cisco Discovery Protocol (CDP) – a protocol used by Cisco devices to advertise themselves to other Cisco devices. Foundry devices use this protocol to learn device and interface information for Cisco devices in the network

Using FDP

FDP enables Foundry devices to advertise themselves to other Foundry devices on the network. When you enable FDP on a Foundry 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 Foundry device running FDP sends FDP updates on Layer 2 to MAC address 01-E0-52-CC-CC-CC. Other Foundry 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 Foundry 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 Foundry 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 Foundry 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 35.1: 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 :
Foundry Networks, 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 35.2: 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 :
Foundry Networks, 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, see Table 35.2 on page 35-4.

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 Foundry 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 36
Remote Network Monitoring

This chapter describes the remote monitoring features available on Foundry products:

- Remote Monitoring (RMON) statistics – All Foundry products support RMON statistics on the individual port level. See “RMON Support” on page 36-2.
- sFlow – sFlow collects interface statistics and traffic samples from individual interfaces on a BigIron RX and exports the information to a monitoring server. See “sFlow” on page 37-1.

Basic Management

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 Foundry 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 36.1: 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.
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.

Table 36.1: Export Configuration and Statistics (Continued)

This Line...	Displays...
Fragments	The total number of packets received that were less than 64 octets long and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). It is normal for this counter to increment, since it counts both runts (which are normal occurrences due to collisions) and noise hits. This number does not include framing bits but does include FCS octets.
Oversize packets	The total number of packets received that were longer than 1518 octets and were otherwise well formed. This number does not include framing bits but does include FCS octets.
Jabbers	The total number of packets received that were longer than 1518 octets and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). **Note**: This definition of jabber is different from the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). These documents define jabber as the condition where any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms. This number does not include framing bits but does include FCS octets.
Collisions	The best estimate of the total number of collisions on this Ethernet segment.
64 octets pkts	The total number of packets received that were 64 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
65 to 127 octets pkts	The total number of packets received that were 65 – 127 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
128 to 255 octets pkts	The total number of packets received that were 128 – 255 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
256 to 511 octets pkts	The total number of packets received that were 256 – 511 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.

Table 36.1: 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.
1024 to 1518 octets pkts	The total number of packets received that were 1024 – 1518 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.

NOTE: The number of entries in a RMON statistics table directly corresponds to the number of ports on a system. For example, if the system is a 26 port device, there will be 26 entries in the statistics display.

History (RMON Group 2)

All active ports by default will generate two history control data entries per active 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: rmon history <entry-number> interface ethernet <slot/port> | management <num> buckets <number> interval <sampling-interval> owner <text-string>

You can modify the sampling interval and the bucket (number of entries saved before overwrite) using the CLI. In the above example, owner refers to the RMON station that will request the information.

NOTE: To review the control data entry for each port or interface, enter the **show rmon history** command.

Alarm (RMON Group 3)

Alarm is designed to monitor configured thresholds for any SNMP integer, time tick, gauge or counter MIB object. Using the CLI, you can define what MIB objects are monitored, the type of thresholds that are monitored (falling, rising or both), the value of those thresholds, and the sample type (absolute or delta).

An alarm event is reported each time that a threshold is exceeded. The alarm entry also indicates the action (event) to be taken if the threshold be exceeded.

A sample CLI alarm entry and its syntax is shown below:

```
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 <threshold-type> can be falling-threshold or rising-threshold.

Event (RMON Group 9)

There are two elements to the Event Group—the ***event control table*** and the ***event log table***.

The event control table defines the action to be taken when an alarm is reported. Defined events can be found by entering the CLI command, show event. The Event Log Table collects and stores reported events for retrieval by an RMON application.

A sample entry and syntax of the event control table is shown below:

```
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 37
sFlow

sFlow is a system for observing traffic flow patterns and quantities within and among a set of BigIron RX devices. To support sFlow:

- Sample packet flows
- Collect the packet headers from sampled packets and collect ingress-egress information on these packets
- Compose the collected information into flow sample messages
- Relay these messages to an external device known as a collector

Participating devices also relay byte and packet counter data (counter samples) for ports to the collector.

sFlow is described in RFC 3176, “InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks”. Refer to this RFC to determine the contents of the sampled packet.

Configuration Considerations

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.

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. See “Enabling sFlow Forwarding” on page 37-4 and “Displaying sFlow Information” on page 37-5.

NOTE: If you change the address sFlow will use for the agent_address, you must disable and re-enable sFlow to enable the feature to use the changed address.

Sampling Rate

The ***sampling rate*** is the average ratio of the number of packets incoming on an sflow enabled port, to the number of flow samples taken from those packets. BigIron RX ports send only the sampled traffic to the CPU. sFlow sampling can affect performance in some configurations, especially if a high sampling rate is implemented.

Port Monitoring

Port monitoring and sFlow are not supported together. If you enable port monitoring on any port, sFlow is disabled for all ports.

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. See "Source Address" on page 37-1.

Changing the Polling Interval

The polling interval defines how often sFlow byte and packet counter data for a port are sent to the sFlow collector(s). 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 (such as 32 in a 15-slot chassis with 232 Gigabit Ethernet ports), 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: Foundry 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. See “Source Address” on page 37-1 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 and/or outbound ports, if that information is available. For information about 802.1X, see “Configuring 802.1X Port Security” on page 32-1.

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

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 37.1: 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. See “Source Address” on page 37-1.
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 collector(s).
sFlow ports	The ports on which you enabled sFlow.
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.

Appendix A
Using Syslog

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, see "Displaying the Syslog Configuration" on page A-3.

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 A.1: CLI Display of Syslog Buffer Configuration

This Field...	Displays...
Syslog logging	The state (enabled or disabled) of the Syslog buffer.

Table A.1: CLI Display of Syslog Buffer Configuration (Continued)

This Field...	Displays...
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. See "Disabling Logging of a Message Level" on page A-8. 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. See "Clearing the Syslog Messages from the Local Buffer" on page A-10.
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.

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/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 A.2 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 A.2: Foundry 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. The location can be one of the following:
		• In 4-slot s:
		  • left side power supply
		  • right side power supply
		• In 8-slot s:
		  • bottom power supply
		  • middle bottom power supply
		  • middle top power supply
		  • top power supply
		• In 15-slot s:
		  • left side power supply
		  • second from left power supply
		  • second from right power supply
		  • right side power supply
		• In Stackable devices:
		  • power supply on right connector
		  • power supply on left connector

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation

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. The location can be one of the following:
		• In Stackable devices:
		  • fan on right connector
		  • fan on left connector
		• In 4-slot s:
		  • left side panel, back fan
		  • left side panel, front fan
		  • rear/back panel, left fan
		  • rear/back panel, right fan
		• In 8-slot and 15-slot s:
		  • rear/back panel, top fan
		  • rear/back panel, bottom fan
		  • top panel, fan

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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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.
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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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.
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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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 Foundry 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 event(s)	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 A.2: Foundry 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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	ACL insufficient L4 session resource, using flow based ACL instead	The device does not have enough Layer 4 session entries. To correct this condition, allocate more memory for sessions. To allocate more memory, enter the following command at the global CONFIG level of the CLI interface: **system-max session-limit** <num>
Notification	ACL 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, see the “Changing CAM Partitions“ chapter in the *Foundry 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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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
Notification	OSPF virtual intf state changed, rid <router-id>, area <area-id>, nbr <ip-addr>, state <ospf-state>	Indicates that the state of an OSPF virtual routing interface has changed. The <router-id> is the router ID of the router the interface is on. The <area-id> is the area the interface is in. The <ip-addr> is the IP address of the OSPF neighbor. The <ospf-state> indicates the state to which the interface has changed and can be one of the following: • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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 A.2: Foundry 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

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF intf config error, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	Indicates that an OSPF interface configuration error has occurred. The <router-id> is the router ID of the 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 A.2: Foundry 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 A.2: Foundry 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 A.2: Foundry 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

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF intf rcvd bad pkt, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, pkt type <pkt-type>	Indicates that an OSPF interface received a bad packet. The <router-id> is the router ID of the 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 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

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF intf retransmit, rid <router-id>, intf addr <ip-addr>, nbr rid <nbr-router-id>, pkt type is <pkt-type>, LSA type <lsa-type>, LSA id <lsa-id>, LSA rid <lsa-router-id>	An OSPF interface on the 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 A.2: Foundry 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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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.
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 Foundry 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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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
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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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.
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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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.
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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Informational	Interface <portnum>, line protocol down	The line protocol on a port has gone down. The <portnum> is the port number.
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.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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.
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)

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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> attempt(s)	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 via 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 via the Web, SNMP, console, SSH, or Telnet session.
Informational	ACL <acl id> added | deleted | modified from console | telnet | ssh | web | snmp session	A user created, modified, deleted, or applied an ACL via 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 via 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 via 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 via 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 via the Web, SNMP, console, SSH, or Telnet session.

Table A.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
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 Foundry Technical Support.

Appendix B
Commands That Require a Reload

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 B.1 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 B.1: Commands That Require a Software Reload

Command	See ...
max-frame-size	"Setting Maximum Frame Size Per PPCR" on page 18-18
multicast-flooding	"Hardware Flooding for Layer 2 Multicast and Broadcast Packets" on page 9-42
system-max	"Displaying and Modifying System Parameter Default Settings" on page 4-13

Index of CLI Commands

S

